



*water*

# Flood Forecasting Using Machine Learning Methods

---

Edited by  
Fi-John Chang, Kuolin Hsu and Li-Chiu Chang  
Printed Edition of the Special Issue Published in *Water*

# **Flood Forecasting Using Machine Learning Methods**



# Flood Forecasting Using Machine Learning Methods

Special Issue Editors

**Fi-John Chang**

**Kuolin Hsu**

**Li-Chiu Chang**

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade



*Special Issue Editors*

Fi-John Chang

National Taiwan University  
Taiwan

Kuolin Hsu

University of California  
USA

Li-Chiu Chang

Tamkang University  
Taiwan

*Editorial Office*

MDPI

St. Alban-Anlage 66  
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Water* (ISSN 2073-4441) from 2018 to 2019 (available at: [https://www.mdpi.com/journal/water/special\\_issues/flood\\_forecast](https://www.mdpi.com/journal/water/special_issues/flood_forecast))

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. *Journal Name Year, Article Number, Page Range.*

**ISBN 978-3-03897-548-9 (Pbk)**

**ISBN 978-3-03897-549-6 (PDF)**

Cover image courtesy of Fi-John Chang.

© 2019 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license, which allows users to download, copy and build upon published articles, as long as the author and publisher are properly credited, which ensures maximum dissemination and a wider impact of our publications.

The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons license CC BY-NC-ND.

# Contents

<b>About the Special Issue Editors</b>	vii
<b>Preface to "Flood Forecasting Using Machine Learning Methods"</b>	ix
<b>Li-Chiu Chang, Fi-John Chang, Shun-Nien Yang, I-Feng Kao, Ying-Yu Ku, Chun-Ling Kuo and Ir. Mohd Zaki bin Mat Amin</b>	
Building an Intelligent Hydroinformatics Integration Platform for Regional Flood Inundation Warning Systems	
Reprinted from: <i>Water</i> 2019, 11, 9, doi:10.3390/w11010009	1
<b>Amir Mosavi, Pinar Ozturk and Kwok-wing Chau</b>	
Flood Prediction Using Machine Learning Models: Literature Review	
Reprinted from: <i>Water</i> 2018, 10, 1536, doi:10.3390/w10111536	14
<b>Yinghao Yu, Hongbo Zhang and Vijay P. Singh</b>	
Forward Prediction of Runoff Data in Data-Scarce Basins with an Improved Ensemble Empirical Mode Decomposition (EEMD) Model	
Reprinted from: <i>Water</i> 2018, 10, 388, doi:10.3390/w10040388	54
<b>Yang Chen, Rongshuang Fan, Xiucheng Yang, Jingxue Wang and Aamir Latif</b>	
Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning	
Reprinted from: <i>Water</i> 2018, 10, 585, doi:10.3390/w10050585	69
<b>Jianzhong Zhou, Tian Peng, Chu Zhang and Na Sun</b>	
Data Pre-Analysis and Ensemble of Various Artificial Neural Networks for Monthly Streamflow Forecasting	
Reprinted from: <i>Water</i> 2018, 10, 628, doi:10.3390/w10050628	89
<b>You-Da Jhong, Chang-Shian Chen, Hsin-Ping Lin and Shien-Tsung Chen</b>	
Physical Hybrid Neural Network Model to Forecast Typhoon Floods	
Reprinted from: <i>Water</i> 2018, 10, 632, doi:10.3390/w10050632	105
<b>Mohammad Ehteram, Faridah Binti Othman, Zaher Mundher Yaseen, Haitham Abdulmohsin Afan, Mohammed Falah Allawi, Marlinda Bt. Abdul Malek, Ali Najah Ahmed, Shamsuddin Shahid, Vijay P. Singh and Ahmed El-Shafie</b>	
Improving the Muskingum Flood Routing Method Using a Hybrid of Particle Swarm Optimization and Bat Algorithm	
Reprinted from: <i>Water</i> 2018, 10, 807, doi:10.3390/w10060807	122
<b>Gokmen Tayfur, Vijay P. Singh, Tommaso Moramarco and Silvia Barbetta</b>	
Flood Hydrograph Prediction Using Machine Learning Methods	
Reprinted from: <i>Water</i> 2018, 10, 968, doi:10.3390/w10080968	143
<b>Saeed Farzin, Vijay P. Singh, Hojat Karami, Nazarin Farahani, Mohammad Ehteram, Ozgur Kisi, Mohammed Falah Allawi, Nuruol Syuhadaa Mohd and Ahmed El-Shafie</b>	
Flood Routing in River Reaches Using a Three-Parameter Muskingum Model Coupled with an Improved Bat Algorithm	
Reprinted from: <i>Water</i> 2018, 10, 1130, doi:10.3390/w10091130	156

<b>Dieu Tien Bui, Khabat Khosravi, Shaojun Li, Himan Shahabi, Mahdi Panahi, Vijay P. Singh, Kamran Chapi, Attaollah Shirzadi, Somayeh Panahi, Wei Chen and Baharin Bin Ahmad</b> New Hybrids of ANFIS with Several Optimization Algorithms for Flood Susceptibility Modeling Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1210, doi:10.3390/w10091210 . . . . .	180
<b>Li-Chiu Chang, Mohd Zaki M. Amin, Shun-Nien Yang and Fi-John Chang</b> Building ANN-Based Regional Multi-Step-Ahead Flood Inundation Forecast Models Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1283, doi:10.3390/w10091283 . . . . .	208
<b>Yen-Ming Chiang, Ruo-Nan Hao, Jian-Quan Zhang, Ying-Tien Lin and Wen-Ping Tsai</b> Identifying the Sensitivity of Ensemble Streamflow Prediction by Artificial Intelligence Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1341, doi:10.3390/w10101341 . . . . .	226
<b>Lu Chen, Na Sun, Chao Zhou, Jianzhong Zhou, Yanlai Zhou, Junhong Zhang and Qing Zhou</b> Flood Forecasting Based on an Improved Extreme Learning Machine Model Combined with the Backtracking Search Optimization Algorithm Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1362, doi:10.3390/w10101362 . . . . .	245
<b>Chen Liang, Hongqing Li, Mingjun Lei and Qingyun Du</b> Dongting Lake Water Level Forecast and Its Relationship with the Three Gorges Dam Based on a Long Short-Term Memory Network Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1389, doi:10.3390/w10101389 . . . . .	262
<b>Tao Bai, Jian Wei, Wangwang Yang and Qiang Huang</b> Multi-Objective Parameter Estimation of Improved Muskingum Model by Wolf Pack Algorithm and Its Application in Upper Hanjiang River, China Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1415, doi:10.3390/w10101415 . . . . .	282
<b>Paul Muñoz, Johanna Orellana-Alvear, Patrcik Willems and Rolando Céllerí</b> Flash-Flood Forecasting in an Andean Mountain Catchment—Development of a Step-Wise Methodology Based on the Random Forest Algorithm Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1519, doi:10.3390/w10111519 . . . . .	296
<b>Caihong Hu, Qiang Wu, Hui Li, Shengqi Jian, Nan Li and Zhengzheng Lou</b> Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1543, doi:10.3390/w10111543 . . . . .	314
<b>Minghong Chen, Juanjuan Pang and Pengxiang Wu</b> Flood Routing Model with Particle Filter-Based Data Assimilation for Flash Flood Forecasting in the Micro-Model of Lower Yellow River, China Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1612, doi:10.3390/w10111612 . . . . .	330
<b>Aida Jabbari and Deg-Hyo Bae</b> Application of Artificial Neural Networks for Accuracy Enhancements of Real-Time Flood Forecasting in the Imjin Basin Reprinted from: <i>Water</i> <b>2018</b> , <i>10</i> , 1626, doi:10.3390/w10111626 . . . . .	345

## About the Special Issue Editors

**Fi-John Chang** (Distinguished Professor) has been engaged for the past thirty years, in conducting researches in technology revolution and the modernization of data-driven modelling, particularly in the fields of artificial intelligence (including machine learning, artificial neural networks, evolutionary algorithms) and interdisciplinary studies involving engineering, hydroinformatics, and environment. Innovative techniques/solutions developed by his team have been applied with success to governmental/NGO projects involving hydro-meteorology, integrated water resources management, eco-hydrosystems, water quantity/quality, etc. More than 200 papers have been published in peer-review journals. His international collaborations have engaged partners from Asia, Europe, and USA. He has long served as Associate Editor/Guest Editor for several prominent hydrological journals like *Water*. He is the Funding President of the Taiwan Hydro-Informatics Society, which aims to develop hydroinformatics sciences and technology-oriented researches.

**Kuolin Hsu** (Professor) is a professor of civil and environmental engineering at the University of California, Irvine. He majored in hydrology and water resources at the University of Arizona. His research interests cover remote sensing of precipitation, hydrological process modeling, water resources systems analysis, and machine learning in hydrological and environmental applications. Dr. Hsu's recent research is focused on precipitation sensing using satellite, radar, and in-situ observations and on modeling of hydrological extreme events. He serves as an associate director of the center for hydrometeorology and remote sensing (CHRS). Dr. Hsu's research has been funded by government agencies, such as NASA, NOAA, NSF, and DOE.

**Li-Chiu Chang** (Professor) has more than 15 years of experience leveraging artificial neural network models, fuzzy inference systems, and data mining to develop innovative methods for hydrosystems. She has coordinated more than 20 government projects related to developing and designing real-time engines for reservoir flood operation and an integration platform for intelligent city flood warning systems in Taiwan. She has published over 60 technical and scientific papers in peer-reviewed journals and co-authored a textbook entitled *Introduction to Artificial Neural Networks: Principles and Applications*. Her current research interests include water resources management, reservoir operation, artificial intelligence (artificial neural networks, genetic algorithms, fuzzy sets), flood forecasting, and data mining.



# Preface to "Flood Forecasting Using Machine Learning Methods"

Nowadays, the degree and scale of flood hazards has been massively increasing as a result of the changing climate, and large-scale floods jeopardize lives and properties, causing great economic losses, in the inundation-prone areas of the world. Early flood warning systems are promising countermeasures against flood hazards and losses. A collaborative assessment according to multiple disciplines, comprising hydrology, remote sensing, and meteorology, of the magnitude and impacts of flood hazards on inundation areas significantly contributes to model the integrity and precision of flood forecasting. Methodologically oriented countermeasures against flood hazards may involve the forecasting of reservoir inflows, river flows, tropical cyclone tracks, and flooding at different lead times and/or scales. Analyses of impacts, risks, uncertainty, resilience, and scenarios coupled with policy-oriented suggestions will give information for flood hazard mitigation. Emerging advances in computing technologies coupled with big-data mining have boosted data-driven applications, among which Machine Learning technology, with its flexibility and scalability in pattern extraction, has modernized not only scientific thinking but also predictive applications. This book explores recent Machine Learning advances in flood forecast and management in a timely manner and presents interdisciplinary approaches to modelling the complexity of flood hazards-related issues, with contributions to integrative solutions from a local, regional, or global perspective.

**Fi-John Chang, Kuolin Hsu, Li-Chiu Chang**  
*Special Issue Editors*



*Editorial*

# Building an Intelligent Hydroinformatics Integration Platform for Regional Flood Inundation Warning Systems

Li-Chiu Chang <sup>1,\*</sup>, Fi-John Chang <sup>2</sup>, Shun-Nien Yang <sup>1</sup>, I-Feng Kao <sup>2</sup>, Ying-Yu Ku <sup>1</sup>, Chun-Ling Kuo <sup>3</sup> and Ir. Mohd Zaki bin Mat Amin <sup>4</sup>

- <sup>1</sup> Department of Water Resources and Environmental Engineering, Tamkang University, New Taipei City 25137, Taiwan; aa22814946@yahoo.com.tw (S.-N.Y.); kuyingyu526@gmail.com (Y.-Y.K.)  
<sup>2</sup> Department of Bioenvironmental Systems Engineering, National Taiwan University, Taipei 10617, Taiwan; changfj@ntu.edu.tw (F.-J.C.); d05622002@ntu.edu.tw (I.-F.K.)  
<sup>3</sup> Water Resources Agency, Ministry of Economic Affairs, Taipei 10617, Taiwan; a680010@wra.gov.tw  
<sup>4</sup> Water Resources and Climatic Change Research Centre, National Hydraulic Research Institute of Malaysia, 43300 Selangor, Malaysia; zaki@nahrim.gov.my  
\* Corresponding author: changlc@mail.tku.edu.tw; Tel.: +886-2-26258523

Received: 30 November 2018; Accepted: 19 December 2018; Published: 21 December 2018

**Abstract:** Flood disasters have had a great impact on city development. Early flood warning systems (EFWS) are promising countermeasures against flood hazards and losses. Machine learning (ML) is the kernel for building a satisfactory EFWS. This paper first summarizes the ML methods proposed in this special issue for flood forecasts and their significant advantages. Then, it develops an intelligent hydroinformatics integration platform (IHIP) to derive a user-friendly web interface system through the state-of-the-art machine learning, visualization and system developing techniques for improving online forecast capability and flood risk management. The holistic framework of the IHIP includes five layers (data access, data integration, servicer, functional subsystem, and end-user application) and one database for effectively dealing with flood disasters. The IHIP provides real-time flood-related data, such as rainfall and multi-step-ahead regional flood inundation maps. The interface of Google Maps fused into the IHIP significantly removes the obstacles for users to access this system, helps communities in making better-informed decisions about the occurrence of floods, and alerts communities in advance. The IHIP has been implemented in the Tainan City of Taiwan as the study case. The modular design and adaptive structure of the IHIP could be applied with similar efforts to other cities of interest for assisting the authorities in flood risk management.

**Keywords:** machine learning; early flood warning systems; hydroinformatics; database; flood forecast; Google Maps

## 1. Introduction

The degree and scale of flood hazards have increased massively with the changing climate in the last decades, and large-scale flash floods bring fast-moving and rapid-rising water with force, resulting in tremendous life and property losses as well as social disruption worldwide. Building resilience to natural disasters is one of the most pressing challenges for achieving sustainable urban development in flood-prone regions. Early flood warning systems (EFWS) are promising countermeasures against flood hazards and losses. The core of a satisfactory EFWS is the capability of sequentially providing a reliable and accurate forecast of the regional flood inundation depths with sufficient lead time. Machine Learning (ML) technologies have boosted mess applications because of its great flexibility and scalability in extracting significant features from complex data structures and providing better

performance and cost-effective solutions. Its popularity dramatically increased among hydrologists. This special issue collects 18 original contributions addressing the state of the art of machine-learning methods, conducting a high level of research and providing practical information on how to implement ML methods for flood forecast along with case studies from different regions of the world. The selected manuscripts presented in this special issue provide more insights into machine-learning methods and strategies for accurate flood forecast. Contributions cover a wide range of topics, including:

1. The application of various ML methods with genetic algorithms and/or particle swarm optimization methods for flood hydrograph predictions [1,2];
2. Improved Ensemble Various Artificial Neural Networks (ANNs) for hydrological series extension and monthly streamflow forecasting [3–5];
3. Development of methodology for improving extreme learning machine method [6], random forest algorithm [7] and adaptive neuro-fuzzy inference system (ANFIS) with optimization algorithms for the forecast [8];
4. Improving the Muskingum Routing method using various optimization methods such as hybrid bat-swarm algorithm [9], improved bat algorithm [10], and Wolf Pack Algorithm [11], or combined with a particle filter-based assimilation model [12] for streamflow forecasts;
5. Using hybrid neural network models (self-organizing map and back-propagation neural networks) to model the rainfall-runoff process for flood forecasts [13];
6. Using Deep Learning (DL), convolutional neural network (CNN), for extracting urban water bodies based on remote-sensing imagery [14], and Long Short-Term Memory Network (LSTM) for the Dongting Lake water level forecast [15] and rainfall-runoff Simulation [16];
7. Building ANN-based methods—the self-organizing map (SOM) and recurrent nonlinear autoregressive with exogenous inputs (RNARX) to sequentially forecast area-wide inundation depths shown on Google Earth [17];
8. Providing an extensive overview of the various ML algorithms used in flood prediction and introducing the most promising prediction methods for both long-term and short-term floods [18].

Information systems play an important role in environmental and geoscience disciplines by providing integrated multi-disciplinary platforms that combine data management, visualization, analysis, modeling, and information communication capabilities. Recently, there has been growing interest in building flood early warning systems (FEWS) for extreme storms in an urban city. This has illustrated the increasing deployment of FEWS due to advances in ensemble weather forecasting, the availability of high-resolution satellite data, and artificial intelligence techniques as well as improved technology for communication and sharing of information. Among the most widely-used tools for online presentation of flood inundation maps with the effectiveness of possible countermeasures, FEWS are now regarded as cost-efficient alternatives to life-saving, damage prevention and the resilience enhancement of a society [19–22]. The provision of flood inundation maps regarding the extents and severity degrees of flood impacts plays a vital role in FEWS, and these maps provide a fundamental building block of ready-to-access content so that decisions can be made upon them. Given sufficient notice in a clear and informative manner, disaster damages can be mitigated considerably. Real-time access, analytical processing, and interactive visualizations are crucial to deliver a better understanding of flood-related issues [23]. Traditional systems that share flood-related data and information, however, have limited capabilities of integrating distributed data as well as visualizing and communicating modeling results. Recent advances in science and technology, especially computer-based data processing capabilities and communication facilities, have enabled longer lead times many steps ahead of flood forecasts along transboundary river basins. Developing early warning systems for the riparian countries can contribute to the mitigation of flood risks and life-saving through effective utilization of these new tools.

Over the last decades, many efforts have been devoted to flood defense strategies, which have focused mainly on non-structural efforts such as flood alerts and warning systems based on flood-depth forecasts [24–26]. Advances in computational techniques and the emergence of new data sources have

provided hydrologists with tools to make management decisions on flood forecasting and flood risk management. Following this track, hydroinformatics has emerged as an essential tool by combining science, technologies and social considerations into a holistic coherent framework to timely deal with collecting, modeling, visualizing, and sharing flood-related information and to improve the applicability and accuracy of flood warnings [27–34]. ML methods are efficient tools for extracting the key information from complex highly dimensional input–output patterns and are widely used in various hydrological problems such as flood forecasts in this special issue [1–18] as well as groundwater and water management issues [35–46]. Recently, technological advances in social media have improved data gathering and dissemination, especially under the development of world-wide-web technologies. The Internet of Things (IoT) is a system of devices that collect data in real time and transfer it through a wireless network to a communication framework of control centers for analyzing the data and providing suitable countermeasures. Recent studies have indicated that the combination of IoT and machine-learning techniques could be beneficial to flood prediction [47,48]. Geographical information systems (GIS) are a user-friendly interface designed to access geographical services such as viewing, understanding, enquiring, interpreting and visualizing data [49–52]. The Google Maps Application Programming Interface (API) provides a programming application interface to integrate Google Maps into websites. A flood forecasting and warning system upon Google Maps can provide relevant meteorological and hydrological information of an event at critical timings, along with operational actions, results, impacts, and lessons learned. Recently, a spatial decision support system incorporated with wireless sensor networks and volunteered geographic information sources has been deployed in the town of São Carlos in Brazil, which has made flood risk management more effective [23]. Based on an ontology-based approach linking environmental models with disaster-related data, a flood disaster management system was explored to offer one-stop flood disaster management, and a subsequent 3D visualization was provided to improve the interpretability of disaster data and the effectiveness of decision-making processes [53]. Effective flood risk management requires updated information to ensure that the correct decisions can be made and that sufficiently accurate forecasts can be provided to promote community confidence so that the community will respond adequately when receiving warnings. The main challenge is to accurately forecast river flow in near real-time and to timely forecast flood peaks for drainage basins using the same technology.

This study aims to develop a user-friendly real-time regional flood forecasting system based on the state-of-the-art data-mining, visualization and system developing techniques such that actions can be efficiently taken to alleviate flood risks endangering residents and properties in inundation prone regions. We present a comprehensive flood warning platform, the intelligent hydroinformatics integration platform (IHIP), that could integrate data, analysis, module, and visualization to provide real-time Google-maps-based information of forecasted flood depths for urban areas in the Tainan City of Taiwan. The web tools and emerging web technologies used in the IHIP system that access rainfall, flood-related data, real-time flood forecasts, flood inundation maps, and interactive visualizations are introduced in the next section. This is followed by the implementation efforts and the description of the Google-maps-based flood depths information system in the study regions. The final section concludes the findings of this study and recommends future work enabled by the implementation of this system.

## 2. System

The purpose of this study is to provide a one-stop web-platform to visualize flood-related data and online regional flood inundation maps. As the main stakeholders of the web-platform are the public and authoritative officers, it is required to provide an interactive and user-friendly interface that requires minimal technical skills to use and to directly handle the visualization and examination of data within the platform, without external application. To achieve this goal, we develop an intelligent hydroinformatics integrated platform (IHIP) that not only implements the latest scientific technologies such as data acquisition, data assimilation and ML modeling for improving forecast capabilities but

also organizes these issues in an integrative manner. The IHIP involves the processes of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. We further develop the online display of flood inundation maps through a friendly web interface to visualize regional flood depths and interact with users. The framework is established based on the architecture that consists of five layers: data access layer; data integration layer; service layer; functional subsystem layer; and end-user application layer (Figure 1). Their main functions are described as follows.

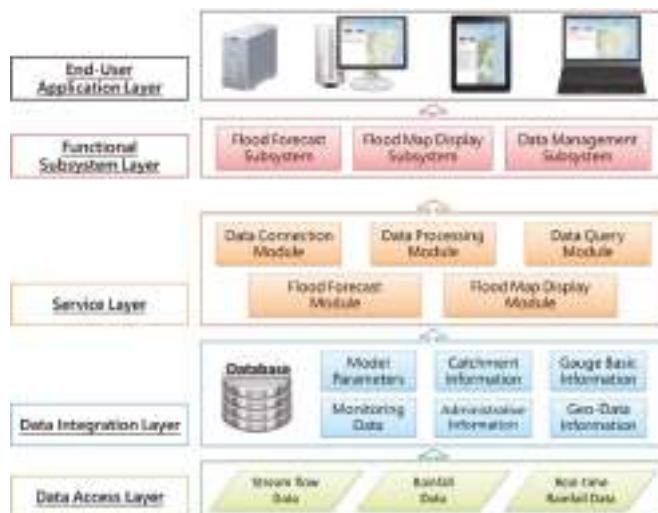


Figure 1. Architecture of the flood early warning system for river basins.

The data access layer provides simplified accesses to the real-time data of the web-database and automatically stores them in the persistent storage of the database. The data integration layer is a relational database and defines data relations which persistently store real-time data, model parameters, as well as the information of catchments, administrative divisions, gauges, and geo-data.

The service layer involves five modules (i.e., data collection, data processing, data query, flood forecast, and flood map display), manages the connections between the modules and database to ensure the effective communication among different layers, and implements the user interface of the end-user application layer.

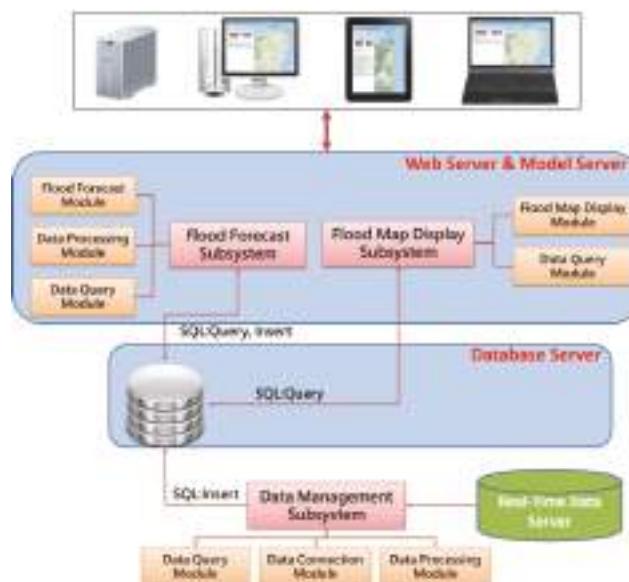
The functional subsystem layer integrates subsystems of flood forecasting, flood map display, and data management not only to perform an important task or specific feature but also to display flood maps derived from the results of flood forecast models.

The end-user application layer is a rich web application interface that handles user interaction and navigation of the data and resources, and it also provides full end-user access to flood forecast results with a visualized geographic map-based website.

The data management, analysis, modeling and visualization of the IHIP provides the core functions of the system, in which the access to data and services from distributed sources is through the use of web services and application interfaces. The IHIP aims at improving the reliability and accuracy of flood forecasts and alleviating the burden imposed on the web server that handles all data queries, data visualization, and data analysis. The next sections will outline the modules, database and data acquisition used in the IHIP in details.

## 2.1. Modules

To integrate the interactions and interchanges among layers and subsystems in the IHIP, several modules are designed. Modular Programming is a technique used to divide the functionality of a program into several independent and interchangeable modules, where each module executes only one aspect of the desired functionality. For interpreting the system, we use the term “module” to separate the independent functions and to explain the interactions and interchanges among modules and subsystems. Here, a module means an object with specific functions supporting different sub-systems when a sub-system interacts with the other part of the whole system. Different sub-systems may require the same functions so that the modules can provide specific and independent functions. This encapsulation is beneficial to the system, for instance, improving maintainability, reducing the impact of a change in data sources on the system, and allowing new data sources to be included in a flexible way. The relationships between four servers (i.e., web, model, database, real-time data), six modules and three sub-systems are shown in Figure 2. The six modules (i.e., Data Connection, Data Query, Data processing, Flood Forecast, Flood Map Display, and Web Server) are explained in more details as follows.



**Figure 2.** Relationships between four servers, five modules and three sub-systems.

**Data Connection Module:** This module builds connections between any kinds of databases, such as Microsoft SQL (Structured Query Language) Server or My SQL. It can build the connection between the system and the database management system, subject to user demands.

**Data Query Module:** This module assists data operators in accessing the Flood Forecast database for information such as query data, store data and forecasted flood data, and to acquire model parameters for building flood forecast models.

**Data Processing Module:** This module conducts data pre-processing, including normalization, transformation, inverse normalization, and missing data treatment.

**Flood Forecast Module:** This module implements a hybrid data-driven model (i.e., the recurrent nonlinear autoregressive with exogenous inputs (RNARX) and the self-organizing map (SOM)) to

build the regional flood inundation forecast model, where those models require automatic execution with forecast results stored into the database of the flood forecasting system.

**Flood Map Display Module:** The Google Maps API of the Google Maps Server Module is used to embed Google Maps into our website, on which site-specific data can be overlaid.

**Web Server Module:** This study focuses on online displaying and organizing regional flood inundation data using up-to-date web technologies driven by the database. Ajax (Asynchronous JavaScript and XML) is a growing web development technique for creating interactive web applications [54,55] and is used in this proposed system. The Web Server Module can also automatically generate the Keyhole Markup Language (KML) files for download purpose. The generated KML files containing placemarks, polygons and textual descriptions can be displayed in Google Earth, ESRI ArcGIS Explorer, Adobe PhotoShop, AutoCAD, and Yahoo Pipes.

## 2.2. Database and Data Acquisition

### 2.2.1. Database

The significant global impact of recurring flooding events on hazard management leads to an increasing demand for comprehensive flood databases. Besides the programming for those modules and servers in the IHIP, a database is a vital part of organizing data and information for building forecasting models, maintaining the relations of the forecasted points, displaying the forecasted results, and managing the models. The relational database is one of the kernel parts for building a real-time visual map-based flood forecast system. This database can be divided into four parts: user management, hydrologic data, model information, and spatial data, respectively, which are explained in more detail as follows.

The user management part maintains the different roles of users, composed of username, password, and authorization. Authorization specifies the access rights to resources related to information security, computer security, and access control. During operation, the system uses these access control rules to decide whether access requests shall be granted or rejected.

The hydrologic data part maintains real-time hydrologic data composed of gauge information and observed gauge data. Real-time rainfall and streamflow data of each gauge have to be stored in this database.

The model information part includes catchment information and model parameters required for re-constructing and re-executing the flood forecast models.

The spatial data part includes administrative division information and geo-data information and provides the information required for flood map display. It provides the forecast models with inputs, maintains the parameters of these forecast models and their relationship, and offers the web-based map display with the geo-data and forecasted flood depths.

Database relationships are defined among real-time data, flood forecast models of different river basins, and geo-data for the map-based flood forecast display.

### 2.2.2. Data Acquisition

One of the most important tasks of the data access layer is to access real-time data because real-time data are the key inputs of the flood forecast models. The Quantitative Precipitation Estimation and Segregation Using Multiple Sensors (QPESUMS) use the Doppler radar, satellite infrared, rain gauge and other data sources to make the Quantitative Precipitation Estimations (QPE) for severe weather systems in Taiwan [56,57]. The Products of QPESUMS are presented in web page format which can be acquired online through retrieving spatial rainfall data when a typhoon (or heavy storm) strikes Taiwan. They are used in the case study of the Tainan City in Taiwan. All programs and related databases are constructed in Microsoft SQL Server 2016, where Visual C# is used to develop four major items: (1) extracting real-time hydrological data; (2) managing real-time data relevant to the models; (3) storing the forecasted data; and (4) providing the information to the visual map-based display.

### 3. System Implementation

With the support from the IHIP, administrators can activate the flood forecast module to obtain real-time multi-step-ahead flood forecasts. The IHIP would integrate the data management module and the forecast module. The forecast module would build the flood forecasting models based on model types and their parameters stored in the flood forecasting system database in the beginning. The data management module would periodically query rainfall data to feed into those flood forecasting models. Then, the forecasting models would periodically compute flood depths, and the forecast module would store these results in the flood forecasting system database. Flood depths are subsequently extracted and integrated with rainfall data for every place in the investigative area at a preset time interval, such as 1 h, during flood events. Users visit the constructed website and choose a specific county to browse the flood forecasting results on Google Maps, with several display levels and zoom levels of Google Maps through the web server module and Google Maps server module services.

The combined application of the GIS and hydrological modeling determines flood extents and analyzes the role of the existing system. The ArcMap 10.0 (ArcGIS) software of the Environmental Systems Research Institute (ESRI) is set for the main operations with the geographical data of the case study area. The necessity of ArcMap 10.0 is high when developing the geometry of the area for further 1D steady flow simulation. For this purpose, the HEC-GeoRAS for ArcMap 10.0 extension is used, whereas the hydraulic modeling and calculations are set to be implemented in the river analysis system software, HEC-RAS (Hydrologic Engineering Center’s River Analysis System), which is the product of US Army Corps of Engineers.

#### 3.1. Study Area

The proposed IHIP system is implemented in the Tainan City of southern Taiwan. Tainan is the oldest city in Taiwan and is known for its history, temples, traditional lifestyles, and traditional snack food. Tainan has a subtropical climate with an annual average rainfall of 2500 mm. The city enjoys a mild and dry climate in winter while rains take place more frequently in summer. Summer is the typhoon season here. Typhoons usually occur between June and November, with the highest number of weather watches coming in August and September. Tainan City has a high population density and is frequently affected by floods. Because of this, flood risk management research on the river catchments in the city is carried out in this study area.

#### 3.2. Online Display

The web server deals merely with sending requests and responses between clients and the IHIP. We design a web page to display the real-time flood inundation forecasting results. The online system display embeds Google Maps in the web page of this flood warning system and provides KMZ (zipped Keyhole Markup Language, for used in Google Earth) files for download purpose. We create the ASP.NET Web application using C# and access data from SQL server using ADO.NET. Considering the diversity of browsers and devices, we adopt the Responsive Web Design (RWD) and choose the layout with three different parts (header, main and footer) of a web page to make our website perform well and look great on any device such as desktops and smartphones.

The online regional flood inundation forecast model can convert a great number of grids' flood depths into regional inundation maps in GIF format and KML format, and thus, the server sends the related files to users without accessing the database. This also enhances server security, reduces the amount of data transmission, makes the website management and maintenance easy, and shortens the waiting time for loading a web page.

Figure 3 shows a web frame of the real-time flood warning system for Tainan City, which consists of three parts: header; main page and footer. The header includes the title of the system, the legend and the KMZ download button for displaying the forecasted results on Google Earth in user devices. The system title and two drop-down lists for years and date times are also displayed. The main page

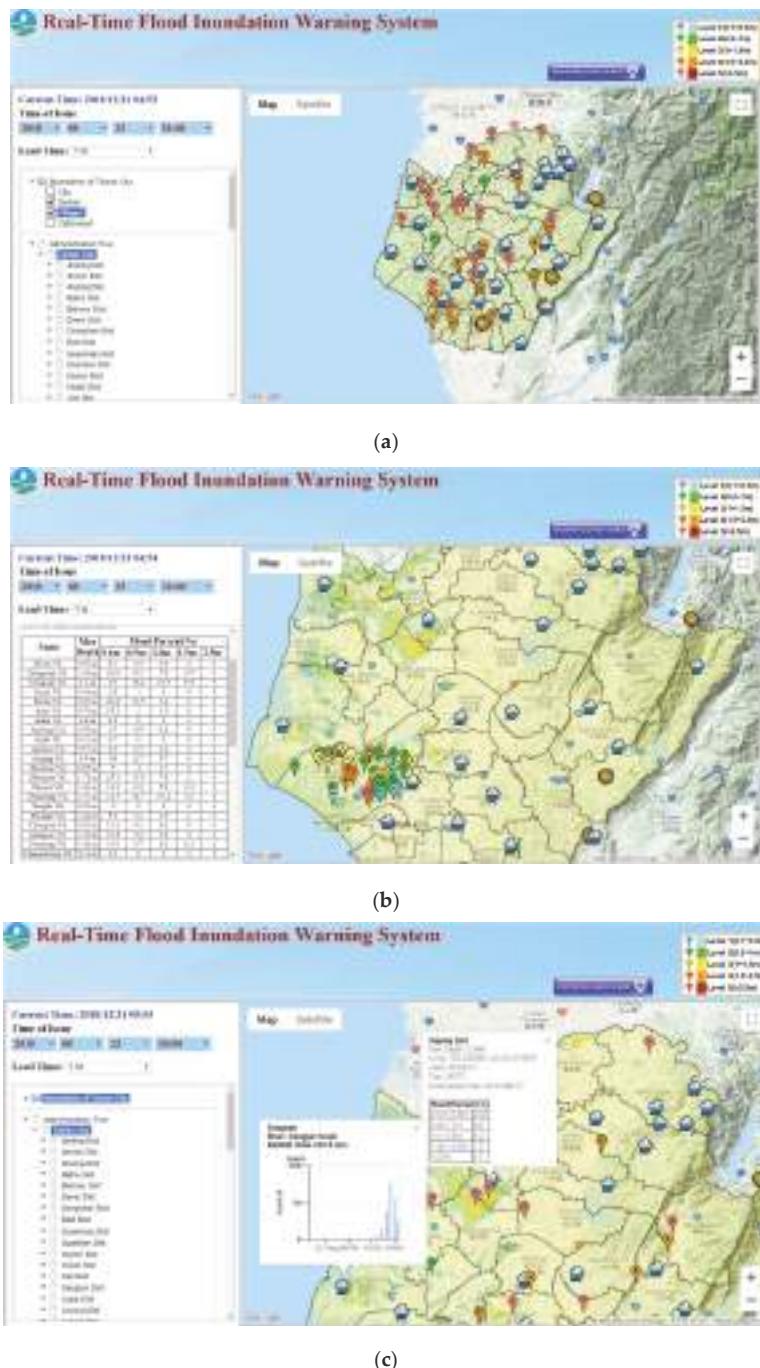
is divided into two parts. The left part includes time of issue, lead time, tree views of boundary and administration for displaying the forecasted results on Google Maps; the remaining space is divided into two sub-parts: control panel and Google Maps. In the control panel, users can access different forecast times, inundated areas, forecast lead times, or read different flood information by selecting an option listed in the drop-down menu. The legend of the flood inundation level is shown in the upper left corner. Here, we provide the quadratic-color flood warning system that shows warning levels (coded by severity) in green, blue, yellow, and red colors such that users can easily visualize (recognize) the inundating depths (conditions) in the whole area. A green warning means level 1 (the least severe) with flood depths ranging from 0.3 to 0.9 m; a blue warning means level 2 with flood depths ranging from 0.9 to 1.5 m; a yellow warning means level 3 with flood depths ranging from 1.5 to 2.5 m; and a red warning means level 4 (the most severe) with flood depths above 2.5 m.



**Figure 3.** A web frame of the real-time flood warning system for Tainan City.

For the right part, click the toggle button of the inundated area to display this area in Google Maps; click the KML download button to download the flood forecast file in KML format; use the forecast time drop-down menu to display the forecasted results at different times; use the drop-down menu to select the administrative area; and use the flood location information drop-down menu to provide the flood depths in a variety of important locations. In the Footer, there are organization logos and copyright information for this online display web page.

Figure 4 shows the presentation screen of the system on the display interface. Figure 4a selects the administrative area (Districts, Villages of the Tainan City); Figure 4b provides the flood depths in selected villages and Figure 4c displays the rainfall and the maximal flood inundation depths of the selected gauges on Google Maps.



**Figure 4.** Main functions of the display interface. (a) Districts, Villages; (b) Table of flood information in selected villages; (c) Rainfall information and the maximal flood inundation depths display on Google Map.

First of all, users can choose the forecast time through the Report Time menu in the top right corner, which is presented by the year, month, date, and hour (ex., 2018/08/23, 20:00), as shown in Figure 4. After time selection, the system will fill in the forecast area, time interval, inundation information, and others. As shown, the maximum inundation depth of each selected village will be tabulated at the bottom left of the interface, and its location will be marked on the map simultaneously. Furthermore, when clicking the Legend icon at the top of the screen, it will display the illustrations of different colors or makers on the map. At last, the KMZ file can be downloaded by clicking the KML download button on Google Earth.

The IHIP allows the monitoring of flooding in a community or a larger region within a single interface and thus, reduces the dependence on tracking multiple sources of information. Second, real-time data from stream gauges and flood forecasting models helps decision-makers and the public take necessary actions to prepare for flood events. Third, the support for multiple devices allows users to access data and information anywhere, and web service availability makes data within the IHIP accessible from other information systems and applications. The IHIP offers a comprehensive information system for improving real-time information management and knowledge transfer pertinent to flood events.

#### 4. Conclusions

A flood early warning system enables an advanced warning of probable flash floods and regional inundation depths for disseminating alarms in flood-threatened areas. A real-time flood inundation map can convey a strong message about the extents and severity degrees of flooding in inundation-prone regions, together with the effectiveness of possible countermeasures. We develop an intelligent hydroinformatics integration platform (IHIP) to provide online forecasting of regional flood depths through the use of the latest hydroinformatics technologies such that actions can be efficiently taken to mitigate flood risks. The holistic framework of the IHIP coupled with the coherent sub-systems, modules, database, and interfaces are introduced. We construct the database and data acquisition under the Microsoft Windows Operating System platform. A web interface is designed to display the forecasts of flood inundation depths in the study areas using the up-to-date web technologies driven by the database. The established flood inundation maps can visualize the potential effects of flooding and communicate scenario development with all stakeholders. The IHIP can be easily executed through the database to simultaneously access the datasets and update the parameters and information obtained from model outputs. The system can very quickly (in just a few seconds) carry out multi-step-ahead forecasts of area-wide inundation maps and thereby lead to real-time flood forecasting. A quadratic-color flood inundation map showing inundated levels in different colors in Google Maps can help users to easily visualize (identify) the threats and impacts of floods on the whole area.

The IHIP is designed in an adaptive and integrated structure that can integrate information from distributed sources and allows an easy development of similar systems for other regions through knowledge transfer. This system can disseminate flood-related data and information to the public such as forecast flood occurrence and alert communities in advance for reducing flood damages.

**Author Contributions:** Conceptualization and Methodology, L.-C.C. and F.-J.C.; Software, L.-C.C., S.-N.Y., I.-F.K., Analysis, L.-C.C., S.-N.Y., Y.-Y.K.; Resources, C.-L.K. and Ir. M.Z. bin M.A.; Writing-Review & Editing, L.-C.C. and F.-J.C.

**Funding:** This research was funded by the Water Resources Agency, Ministry of Economic Affairs, Taiwan, R.O.C. (Grant number: MOEAWRA1070095).

**Acknowledgments:** The authors gratefully acknowledge the Water Resources Agency, Taiwan for supporting this study. Simulated flood inundation datasets provided by the NCKU Research and Development Foundation and the National Center for High-performance Computing are greatly appreciated. The authors of this editorial and Guest Editor of this special issue would like to thank all authors for their notable contributions to this special issue, the reviewers for devoting their efforts to reviewing the manuscripts, and the Water Editorial team for their great support during the review of the submitted manuscripts.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Tayfur, G.; Singh, V.; Moramarco, T.; Barbetta, S. Flood hydrograph prediction using machine learning methods. *Water* **2018**, *10*, 968. [[CrossRef](#)]
- Jabbari, A.; Bae, D.H. Application of Artificial Neural Networks for Accuracy Enhancements of Real-Time Flood Forecasting in the Imjin Basin. *Water* **2018**, *10*, 1626. [[CrossRef](#)]
- Yu, Y.; Zhang, H.; Singh, V.P. Forward prediction of runoff data in data-scarce basins with an improved ensemble empirical mode decomposition (EEMD) model. *Water* **2018**, *10*, 388. [[CrossRef](#)]
- Zhou, J.; Peng, T.; Zhang, C.; Sun, N. Data Pre-Analysis and Ensemble of Various Artificial Neural Networks for Monthly Streamflow Forecasting. *Water* **2018**, *10*, 628. [[CrossRef](#)]
- Chiang, Y.M.; Hao, R.N.; Zhang, J.Q.; Lin, Y.T.; Tsai, W.P. Identifying the Sensitivity of Ensemble Streamflow Prediction by Artificial Intelligence. *Water* **2018**, *10*, 1341. [[CrossRef](#)]
- Chen, L.; Sun, N.; Zhou, C.; Zhou, J.; Zhou, Y.; Zhang, J.; Zhou, Q. Flood Forecasting Based on an Improved Extreme Learning Machine Model Combined with the Backtracking Search Optimization Algorithm. *Water* **2018**, *10*, 1362. [[CrossRef](#)]
- Muñoz, P.; Orellana-Alvear, J.; Willems, P.; Céller, R. Flash-Flood Forecasting in an Andean Mountain Catchment—Development of a Step-Wise Methodology Based on the Random Forest Algorithm. *Water* **2018**, *10*, 1519. [[CrossRef](#)]
- Tien Bui, D.; Khosravi, K.; Li, S.; Shahabi, H.; Panahi, M.; Singh, V.; Chapi, K.; Shirzadi, A.; Panahi, S.; Chen, W.; et al. New hybrids of anfis with several optimization algorithms for flood susceptibility modeling. *Water* **2018**, *10*, 1210. [[CrossRef](#)]
- Ehteram, M.; Binti Othman, F.; Mundher Yaseen, Z.; Abdulmohsin Afan, H.; Falah Allawi, M.; Najah Ahmed, A.; Shahid, S.; P Singh, V.; El-Shafie, A. Improving the Muskingum flood routing method using a hybrid of particle swarm optimization and bat algorithm. *Water* **2018**, *10*, 807. [[CrossRef](#)]
- Farzin, S.; Singh, V.; Karami, H.; Farahani, N.; Ehteram, M.; Kisi, O.; Allawi, M.; Mohd, N.; El-Shafie, A. Flood routing in river reaches using a three-parameter Muskingum model coupled with an improved bat algorithm. *Water* **2018**, *10*, 1130. [[CrossRef](#)]
- Bai, T.; Wei, J.; Yang, W.; Huang, Q. Multi-Objective Parameter Estimation of Improved Muskingum Model by Wolf Pack Algorithm and Its Application in Upper Hanjiang River, China. *Water* **2018**, *10*, 1415. [[CrossRef](#)]
- Chen, M.; Pang, J.; Wu, P. Flood Routing Model with Particle Filter-Based Data Assimilation for Flash Flood Forecasting in the Micro-Model of Lower Yellow River, China. *Water* **2018**, *10*, 1612. [[CrossRef](#)]
- Jhong, Y.D.; Chen, C.S.; Lin, H.P.; Chen, S.T. Physical Hybrid Neural Network Model to Forecast Typhoon Floods. *Water* **2018**, *10*, 632. [[CrossRef](#)]
- Chen, Y.; Fan, R.; Yang, X.; Wang, J.; Latif, A. Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning. *Water* **2018**, *10*, 585. [[CrossRef](#)]
- Liang, C.; Li, H.; Lei, M.; Du, Q. Dongting Lake Water Level Forecast and Its Relationship with the Three Gorges Dam Based on a Long Short-Term Memory Network. *Water* **2018**, *10*, 1389. [[CrossRef](#)]
- Hu, C.; Wu, Q.; Li, H.; Jian, S.; Li, N.; Lou, Z. Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *Water* **2018**, *10*, 1543. [[CrossRef](#)]
- Chang, L.C.; Amin, M.; Yang, S.N.; Chang, F.J. Building ANN-Based Regional Multi-Step-Ahead Flood Inundation Forecast Models. *Water* **2018**, *10*, 1283. [[CrossRef](#)]
- Mosavi, A.; Ozturk, P.; Chau, K.W. Flood Prediction Using Machine Learning Models: Literature Review. *Water* **2018**, *10*, 1536. [[CrossRef](#)]
- Alfieri, L.; Thielen, J. A European precipitation index for extreme rain-storm and flash flood early warning. *Meteorol. Appl.* **2015**, *22*, 3–13. [[CrossRef](#)]
- Pappenberger, F.; Cloke, H.L.; Parker, D.J.; Wetterhall, F.; Richardson, D.S.; Thielen, J. The monetary benefit of early flood warnings in Europe. *Environ. Sci. Policy* **2015**, *51*, 278–291. [[CrossRef](#)]
- Cools, J.; Innocenti, D.; O'Brien, S. Lessons from flood early warning systems. *Environ. Sci. Policy* **2016**, *58*, 117–122. [[CrossRef](#)]
- Girons Lopez, M.; Di Baldassarre, G.; Seibert, J. Impact of social preparedness on flood early warning systems. *Water Resour. Res.* **2017**, *53*, 522–534. [[CrossRef](#)]

23. Horita, F.E.; de Albuquerque, J.P.; Degrossi, L.C.; Mendiondo, E.M.; Ueyama, J. Development of a spatial decision support system for flood risk management in Brazil that combines volunteered geographic information with wireless sensor networks. *Comput. Geosci.* **2015**, *80*, 84–94. [[CrossRef](#)]
24. Chan, N.W. Impacts of disasters and disaster risk management in Malaysia: The case of floods. In *Resilience and Recovery in Asian Disasters*; Springer: Tokyo, Japan, 2015; pp. 239–265.
25. Chang, L.C.; Shen, H.Y.; Chang, F.J. Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. *J. Hydrol.* **2014**, *519*, 476–489. [[CrossRef](#)]
26. Chang, L.C.; Shen, H.Y.; Wang, Y.F.; Huang, J.Y.; Lin, Y.T. Clustering-based hybrid inundation model for forecasting flood inundation depths. *J. Hydrol.* **2010**, *385*, 257–268. [[CrossRef](#)]
27. Bogner, K.; Liechti, K.; Zappa, M. Error Correcting and Combining Multi-model Flood Forecasting Systems. In *Advances in Hydroinformatics*; Springer: Singapore, 2018; pp. 569–578.
28. Herrera, M.; Meniconi, S.; Alvisi, S.; Izquierdo, J. *Advanced Hydroinformatic Techniques for the Simulation and Analysis of Water Supply and Distribution Systems*; MDPI: Basel, Switzerland, 2018.
29. Lynggaard-Jensen, A.; Mark, O.; Gourbesville, P. ICT for urban water infrastructure. Available online: [https://www.researchgate.net/publication/295401876\\_ICT\\_for\\_urban\\_water\\_infrastructure](https://www.researchgate.net/publication/295401876_ICT_for_urban_water_infrastructure) (accessed on 19 December 2018).
30. Nguyen, P.; Thorstensen, A.; Sorooshian, S.; Hsu, K.; AghaKouchak, A. Flood forecasting and inundation mapping using HiResFlood-UCI and near-real-time satellite precipitation data: The 2008 Iowa flood. *J. Hydrometeorol.* **2015**, *16*, 1171–1183. [[CrossRef](#)]
31. Pinho, J.; Ferreira, R.; Vieira, L.; Schwanenberg, D. Comparison between two hydrodynamic models for flooding simulations at river Lima basin. *Water Resour. Manag.* **2015**, *29*, 431–444. [[CrossRef](#)]
32. Vojinovic, Z.; Abbott, M.B. Twenty-Five Years of Hydroinformatics. *Water* **2017**, *9*, 59. [[CrossRef](#)]
33. Wan, Z.; Hong, Y.; Khan, S.; Gourley, J.; Flamig, Z.; Kirschbaum, D.; Tang, G. A cloud-based global flood disaster community cyber-infrastructure: Development and demonstration. *Environ. Model. Softw.* **2014**, *58*, 86–94. [[CrossRef](#)]
34. Yan, K.; Di Baldassarre, G.; Solomatine, D.P.; Schumann, G.J.P. A review of low-cost space-borne data for flood modelling: Topography, flood extent and water level. *Hydrolog. Processes* **2015**, *29*, 3368–3387. [[CrossRef](#)]
35. Antonopoulos, V.Z.; Giannouli, S.K.; Antonopoulos, A.V. Artificial neural networks and empirical equations to estimate daily evaporation: Application to lake Vegoritis, Greece. *Hydrolog. Sci. J.* **2016**, *61*, 2590–2599. [[CrossRef](#)]
36. Atieh, M.; Taylor, G.; Sattar, A.M.; Gharabaghi, B. Prediction of flow duration curves for ungauged basins. *J. Hydrol.* **2017**, *545*, 383–394. [[CrossRef](#)]
37. Chang, L.C.; Chen, P.A.; Chang, F.J. Reinforced two-step-ahead weight adjustment technique for online training of recurrent neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1269–1278. [[CrossRef](#)]
38. Chang, F.J.; Chang, L.C.; Huang, C.W.; Kao, I.F. Prediction of monthly regional groundwater levels through hybrid soft-computing techniques. *J. Hydrol.* **2016**, *541*, 965–976. [[CrossRef](#)]
39. Chang, F.J.; Huang, C.W.; Cheng, S.T.; Chang, L.C. Conservation of groundwater from over-exploitation—Scientific analyses for groundwater resources management. *Sci. Total Environ.* **2017**, *598*, 828–838. [[CrossRef](#)]
40. Chang, F.J.; Tsai, M.J. A nonlinear spatio-temporal lumping of radar rainfall for modelling multi-step-ahead inflow forecasts by data-driven techniques. *J. Hydrol.* **2016**, *30*, 1395–1413.
41. Chen, I.T.; Chang, L.C.; Chang, F.J. Exploring the Spatio-Temporal Interrelation between Groundwater and Surface Water by Using the Self-Organizing Maps. *J. Hydrol.* **2018**, *556*, 131–142. [[CrossRef](#)]
42. Mohanty, S.; Jha, M.K.; Raul, S.K.; Panda, R.K.; Sudheer, K.P. Using artificial neural network approach for simultaneous forecasting of weekly groundwater levels at multiple sites. *Water Resour. Manag.* **2015**, *29*, 5521–5532. [[CrossRef](#)]
43. Pektaş, A.O.; Cigizoglu, H.K. Investigating the extrapolation performance of neural network models in suspended sediment data. *Hydrolog. Sci. J.* **2017**, *62*, 1694–1703. [[CrossRef](#)]
44. Shiri, J.; Martí, P.; Nazemi, A.H.; Sadreddini, A.A.; Kisi, O.; Landeras, G.; Fard, A.F. Local vs. external training of neuro-fuzzy and neural networks models for estimating reference evapotranspiration assessed through k-fold testing. *Hydrolog. Res.* **2015**, *46*, 72–88. [[CrossRef](#)]
45. Wang, J.; Shi, P.; Jiang, P.; Hu, J.; Qu, S.; Chen, X.; Chen, Y.; Dai, Y.; Xiao, Z. Application of BP Neural Network Algorithm in Traditional Hydrological Model for Flood Forecasting. *Water* **2017**, *9*, 48. [[CrossRef](#)]

46. Yaseen, Z.M.; Fu, M.; Wang, C.; Mohtar, W.H.M.W.; Deo, R.C.; El-shafie, A. Application of the Hybrid Artificial Neural Network Coupled with Rolling Mechanism and Grey Model Algorithms for Streamflow Forecasting Over Multiple Time Horizons. *Water Resour. Manag.* **2018**, *32*, 1883–1899. [[CrossRef](#)]
47. Balis, B.; Bubak, M.; Harezlak, D.; Nowakowski, P.; Pawlik, M.; Wilk, B. Towards an operational database for real-time environmental monitoring and early warning systems. *Procedia Comput. Sci.* **2017**, *108*, 2250–2259. [[CrossRef](#)]
48. Mitra, P.; Ray, R.; Chatterjee, R.; Basu, R.; Saha, P.; Raha, S.; Barman, R.; Patra, S.; Biswas, S.S.; Saha, S. Flood forecasting using Internet of things and artificial neural networks. In Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 13–15 October 2016; pp. 1–5.
49. Bonilla Valverde, J.P.; Blank, C.; Roidt, M.; Schneider, L.; Stefan, C. Application of a GIS Multi-Criteria Decision Analysis for the Identification of Intrinsic Suitable Sites in Costa Rica for the Application of Managed Aquifer Recharge (MAR) through Spreading Methods. *Water* **2016**, *8*, 391. [[CrossRef](#)]
50. Fernández, P.; Santana, J.; Sánchez, A.; Trujillo, A.; Domínguez, C.; Suárez, J.P. A GIS Water Management System Using Free and Open Source Software. In *Ubiquitous Computing and Ambient Intelligence*; Springer: Cham, Switzerland, 2016; pp. 383–394.
51. Şener, Ş.; Şener, E.; Davraz, A. Evaluation of water quality using water quality index (WQI) method and GIS in Aksu River (SW-Turkey). *Sci. Total Environ.* **2017**, *584*, 131–144. [[CrossRef](#)]
52. Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in GIS. *J. Hydrol.* **2014**, *512*, 332–343. [[CrossRef](#)]
53. Qiu, L.; Du, Z.; Zhu, Q.; Fan, Y. An integrated flood management system based on linking environmental models and disaster-related data. *Environ. Model. Softw.* **2017**, *91*, 111–126. [[CrossRef](#)]
54. Garrett, J.J. Ajax: A New Approach to Web Applications. 2005. Available online: <http://www.adaptivepath.com/publications/essays/archives/000385> (accessed on 19 December 2018).
55. Mahemoff, M. *AJAX Design Patterns: Creating Web 2.0 Sites with Programming and Usability patterns*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2006.
56. Chiou, P.T.K.; Chen, C.R.; Chang, P.L.; Jian, G.J. Status and outlook of very short range forecasting system in Central Weather Bureau, Taiwan. In *Applications with Weather Satellites II*; International Society for Optics and Photonics: Bellingham, WA, USA, 2005; Volume 5658, pp. 185–197.
57. Chen, C.Y.; Lin, L.Y.; Yu, F.C.; Lee, C.S.; Tseng, C.C.; Wang, A.H.; Cheung, K.W. Improving debris flow monitoring in Taiwan by using high-resolution rainfall products from QPESUMS. *Nat. Hazards* **2007**, *40*, 447–461. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Review

# Flood Prediction Using Machine Learning Models: Literature Review

Amir Mosavi <sup>1,\*</sup>, Pinar Ozturk <sup>1,\*</sup> and Kwok-wing Chau <sup>2</sup>

<sup>1</sup> Department of Computer Science (IDI), Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway

<sup>2</sup> Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, Hong Kong, China; dr.kwok-wing.chau@polyu.edu.hk

\* Correspondence: amir.mosavi@ntnu.no (A.M.); pinar@ntnu.no (P.O.)

Received: 1 September 2018; Accepted: 17 October 2018; Published: 27 October 2018

**Abstract:** Floods are among the most destructive natural disasters, which are highly complex to model. The research on the advancement of flood prediction models contributed to risk reduction, policy suggestion, minimization of the loss of human life, and reduction of the property damage associated with floods. To mimic the complex mathematical expressions of physical processes of floods, during the past two decades, machine learning (ML) methods contributed highly in the advancement of prediction systems providing better performance and cost-effective solutions. Due to the vast benefits and potential of ML, its popularity dramatically increased among hydrologists. Researchers through introducing novel ML methods and hybridizing of the existing ones aim at discovering more accurate and efficient prediction models. The main contribution of this paper is to demonstrate the state of the art of ML models in flood prediction and to give insight into the most suitable models. In this paper, the literature where ML models were benchmarked through a qualitative analysis of robustness, accuracy, effectiveness, and speed are particularly investigated to provide an extensive overview on the various ML algorithms used in the field. The performance comparison of ML models presents an in-depth understanding of the different techniques within the framework of a comprehensive evaluation and discussion. As a result, this paper introduces the most promising prediction methods for both long-term and short-term floods. Furthermore, the major trends in improving the quality of the flood prediction models are investigated. Among them, hybridization, data decomposition, algorithm ensemble, and model optimization are reported as the most effective strategies for the improvement of ML methods. This survey can be used as a guideline for hydrologists as well as climate scientists in choosing the proper ML method according to the prediction task.

**Keywords:** flood prediction; flood forecasting; hydrologic model; rainfall-runoff, hybrid & ensemble machine learning; artificial neural network; support vector machine; natural hazards & disasters; adaptive neuro-fuzzy inference system (ANFIS); decision tree; survey; classification and regression trees (CART), data science; big data; artificial intelligence; soft computing; extreme event management; time series prediction

## 1. Introduction

Among the natural disasters, floods are the most destructive, causing massive damage to human life, infrastructure, agriculture, and the socioeconomic system. Governments, therefore, are under pressure to develop reliable and accurate maps of flood risk areas and further plan for sustainable flood risk management focusing on prevention, protection, and preparedness [1]. Flood prediction models are of significant importance for hazard assessment and extreme event management. Robust and accurate prediction contribute highly to water recourse management strategies, policy suggestions and

analysis, and further evacuation modeling [2]. Thus, the importance of advanced systems for short-term and long-term prediction for flood and other hydrological events is strongly emphasized to alleviate damage [3]. However, the prediction of flood lead time and occurrence location is fundamentally complex due to the dynamic nature of climate condition. Therefore, today's major flood prediction models are mainly data-specific and involve various simplified assumptions [4]. Thus, to mimic the complex mathematical expressions of physical processes and basin behavior, such models benefit from specific techniques e.g., event-driven, empirical black box, lumped and distributed, stochastic, deterministic, continuous, and hybrids [5].

Physically based models [6] were long used to predict hydrological events, such as storm [7,8], rainfall/runoff [9,10], shallow water condition [11], hydraulic models of flow [12,13], and further global circulation phenomena [14], including the coupled effects of atmosphere, ocean, and floods [15]. Although physical models showed great capabilities for predicting a diverse range of flooding scenarios, they often require various types of hydro-geomorphological monitoring datasets, requiring intensive computation, which prohibits short-term prediction [16]. Furthermore, as stated in Reference [17], the development of physically based models often requires in-depth knowledge and expertise regarding hydrological parameters, reported to be highly challenging. Moreover, numerous studies suggest that there is a gap in short-term prediction capability of physical models (Costabile and Macchione [15]). For instance, on many occasions, such models failed to predict properly [18]. Van den Honert and McAneney [18] documented the failure in the prediction of floods accrued in Queensland, Australia in 2010. Similarly, numerical prediction models [19] were reported in the advancement of deterministic calculations, and were not reliable due to systematic errors [20]. Nevertheless, major improvements in physically based models of flood were recently reported through the hybridization of models [21], as well as advanced flow simulations [22,23].

In addition to numerical and physical models, data-driven models also have a long tradition in flood modeling, which recently gained more popularity. Data-driven methods of prediction assimilate the measured climate indices and hydro-meteorological parameters to provide better insight. Among them, statistical models of autoregressive moving average (ARMA) [24], multiple linear regression (MLR) [25], and autoregressive integrated moving average (ARIMA) [26] are the most common flood frequency analysis (FFA) methods for modeling flood prediction. FFA was among the early statistical methods for predicting floods [27]. Regional flood frequency analyses (RFFA) [28], more advanced versions, were reported to be more efficient when compared to physical models considering computation cost and generalization. Assuming floods as stochastic processes, they can be predicted using certain probability distributions from historical streamflow data [29]. For instance, the climatology average method (CLIM) [28], empirical orthogonal function (EOF) [30], multiple linear regressions (MLR), quantile regression techniques (QRT) [31], and Bayesian forecasting models [32] are widely used for predicting major floods. However, they were reported to be unsuitable for short-term prediction, and, in this context, they need major improvement due to the lack of accuracy, complexity of the usage, computation cost, and robustness of the method. Furthermore, for reliable long-term prediction, at least, a decade of data from measurement gauges should be analyzed for a meaningful forecast [32]. In the absence of such a dataset, however, FFA can be done using hydrologic models of RFFA, e.g., MISBA [33] and Sacramento [34], as reliable empirical methods with regional applications, where streamflow measurements are unavailable. In this context, distributed numerical models are used as an attractive solution [35]. Nevertheless, they do not provide quantitative flood predictions, and their forecast skill level is "only moderate" and they lack accuracy [36].

The drawbacks of the physically based and statistical models mentioned above encourage the usage of advanced data-driven models, e.g., machine learning (ML). A further reason for the popularity of such models is that they can numerically formulate the flood nonlinearity, solely based on historical data without requiring knowledge about the underlying physical processes. Data-driven prediction models using ML are promising tools as they are quicker to develop with minimal inputs. ML is a field of artificial intelligence (AI) used to induce regularities and patterns, providing easier implementation

with low computation cost, as well as fast training, validation, testing, and evaluation, with high performance compared to physical models, and relatively less complexity [37]. The continuous advancement of ML methods over the last two decades demonstrated their suitability for flood forecasting with an acceptable rate of outperforming conventional approaches [38]. A recent investigation by Reference [39], which compared performance of a number of physical and ML prediction models, showed a higher accuracy of ML models. Furthermore, the literature includes numerous successful experiments of quantitative precipitation forecasting (QPF) using ML methods for different lead-time predictions [40,41]. In comparison to traditional statistical models, ML models were used for prediction with greater accuracy [42]. Ortiz-García et al. [43] described how ML techniques could efficiently model complex hydrological systems such as floods. Many ML algorithms, e.g., artificial neural networks (ANNs) [44], neuro-fuzzy [45,46], support vector machine (SVM) [47], and support vector regression (SVR) [48,49], were reported as effective for both short-term and long-term flood forecast. In addition, it was shown that the performance of ML could be improved through hybridization with other ML methods, soft computing techniques, numerical simulations, and/or physical models. Such applications provided more robust and efficient models that can effectively learn complex flood systems in an adaptive manner. Although the literature includes numerous evaluation performance analyses of individual ML models [49–52], there is no definite conclusion reported with regards to which models function better in certain applications. In fact, the literature includes only a limited number of surveys on specific ML methods in specific hydrology fields [53–55]. Consequently, there is a research gap for a comprehensive literature review in the general applications of ML in all flood resource variables from the perspective of ML modeling and data-driven prediction systems.

Nonetheless, ML algorithms have important characteristics that need to be carefully taken into consideration. The first is that they are as good as their training, whereby the system learns the target task based on past data. If the data is scarce or does not cover varieties of the task, their learning falls short, and hence, they cannot perform well when they are put into work. Therefore, using robust data enrichment is essential through, e.g., implementing a distribution function of sums of weights [56], invariance assessments to retain the group characteristics [57], or recovering the missing variables using causally dependent coefficients [58].

The second aspect is the capability of each ML algorithm, which may vary across different types of tasks. This can also be called a “generalization problem”, which indicates how well the trained system can predict cases it was not trained for, i.e., whether it can predict beyond the range of the training dataset. For example, some algorithms may perform well for short-term predictions, but not for long-term predictions. These characteristics of the algorithms need to be clarified with respect to the type and amount of available training data, and the type of prediction task, e.g., water level and streamflow. In this review, we look into examples of the use of various ML algorithms for various types of tasks. At the abstract level, we decided to divide the target tasks into short-term and long-term prediction. We then reviewed ML applications for flood-related tasks, where we structured ML methods as single methods and hybrid methods. Hybrid methods are those that combine more than one ML method.

Here, we should note that this paper surveys ML models used for predictions of floods on sites where rain gauges or intelligent sensing systems used. Our goal was to survey prediction models with various lead times to floods at a particular site. From this perspective, spatial flood prediction was not involved in this study, as we did not study prediction models used to estimate/identify the location of floods. In fact, we were concerned only with the lead time for an identified site.

## 2. Method and Outline

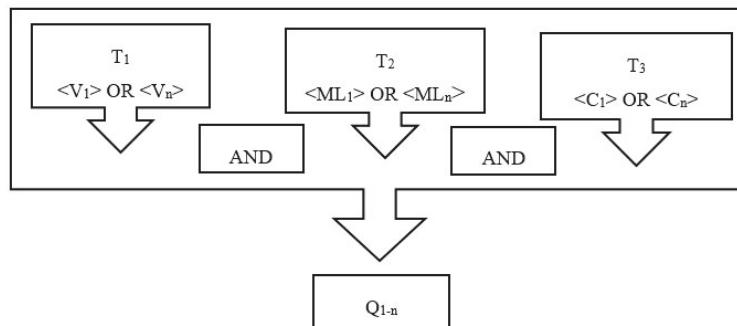
This survey identifies the state of the art of ML methods for flood prediction where peer-reviewed articles in top-level subject fields are reviewed. Among the articles identified, through search queries using the search strategy, those including the performance evaluation and comparison of ML methods were given priority to be included in the review to identify the ML methods that perform better

in particular applications. Furthermore, to choose an article, four types of quality measure for each article were considered, i.e., source normalized impact per paper (SNIP), CiteScore, SCImago journal rank (SJR), and h-index. The papers were reviewed in terms of flood resource variables, ML methods, prediction type, and the obtained results.

The applications in flood prediction can be classified according to flood resource variables, i.e., water level, river flood, soil moisture, rainfall–discharge, precipitation, river inflow, peak flow, river flow, rainfall–runoff, flash flood, rainfall, streamflow, seasonal stream flow, flood peak discharge, urban flood, plain flood, groundwater level, rainfall stage, flood frequency analysis, flood quantiles, surge level, extreme flow, storm surge, typhoon rainfall, and daily flows [59]. Among these key influencing flood resource variables, rainfall and the spatial examination of the hydrologic cycle had the most remarkable role in runoff and flood modeling [60]. This is the reason why quantitative rainfall prediction, including avalanches, slush flow, and melting snow, is traditionally used for flood prediction, especially in the prediction of flash floods or short-term flood prediction [61]. However, rainfall prediction was shown to be inadequate for accurate flood prediction. For instance, the prediction of streamflow in a long-term flood prediction scenario depends on soil moisture estimates in a catchment, in addition to rainfall [62]. Although, high-resolution precipitation forecasting is essential, other flood resource variables were considered in the [63]. Thus, the methodology of this literature review aims to include the most effective flood resource variables in the search queries.

A combination of these flood resource variables and ML methods was used to implement the complete list of search queries. Note that the ML methods for flood prediction may vary significantly according to the application, dataset, and prediction type. For instance, ML methods used for short-term water level prediction are significantly different from those used for long-term streamflow prediction. Figure 1 represents the organization of the search queries and further describes the survey search methodology.

The search query included three main search terms. The flood resource variables were considered as term 1 of the search ( $\langle\text{Flood resource variable1-n}\rangle$ ), which included 25 keywords for search queries mentioned above. Term 2 of search ( $\langle\text{ML method1-m}\rangle$ ) included the ML algorithms. The collection of the references [16,26,28,37,38,42,44] provides a complete list of ML methods, from which the 25 most popular algorithms in engineering applications were used as the keywords of this search. Term 3 included the four search terms most often used in describing flood prediction, i.e., “prediction”, “estimation”, “forecast”, or “analysis”. The total search resulted in 6596 articles. Among them, 180 original research papers were refined through our quality measure included in the survey.



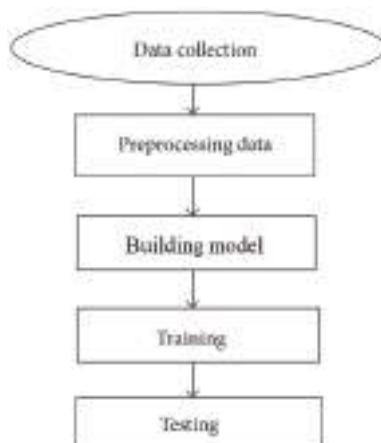
**Figure 1.** Flowchart of the search queries.

Section 3 presents the state of the art of ML in flood prediction. A technical description on the ML method and a brief background in flood applications are provided. Section 4 presents the survey of

ML methods used for short-term flood prediction. Section 5 presents the survey of ML methods used for long-term flood prediction. Section 6 presents the conclusions.

### 3. State of the Art of ML Methods in Flood Prediction

For creating the ML prediction model, the historical records of flood events, in addition to real-time cumulative data of a number of rain gauges or other sensing devices for various return periods, are often used. The sources of the dataset are traditionally rainfall and water level, measured either by ground rain gauges, or relatively new remote-sensing technologies such as satellites, multisensor systems, and/or radars [62]. Nevertheless, remote sensing is an attractive tool for capturing higher-resolution data in real time. In addition, the high resolution of weather radar observations often provides a more reliable dataset compared to rain gauges [63]. Thus, building a prediction model based on a radar rainfall dataset was reported to provide higher accuracy in general [64]. Whether using a radar-based dataset or ground gauges to create a prediction model, the historical dataset of hourly, daily, and/or monthly values is divided into individual sets to construct and evaluate the learning models. To do so, the individual sets of data undergo training, validation, verification, and testing. The principle behind the ML modeling workflow and the strategy for flood modeling are described in detail in the literature [48,65]. Figure 2 represents the basic flow for building an ML model. The major ML algorithms applied to flood prediction include ANNs [66], neuro-fuzzy [67], adaptive neuro-fuzzy inference systems (ANFIS) [68], support vector machines (SVM) [69], wavelet neural networks (WNN) [70], and multilayer perceptron (MLP) [71]. In the following subsections, a brief description and background of these fundamental ML algorithms are presented.



**Figure 2.** Basic flow for building the machine learning (ML) model.

#### 3.1. Artificial Neural Networks (ANNs)

ANNs are efficient mathematical modeling systems with efficient parallel processing, enabling them to mimic the biological neural network using inter-connected neuron units. Among all ML methods, ANNs are the most popular learning algorithms, known to be versatile and efficient in modeling complex flood processes with a high fault tolerance and accurate approximation [39]. In comparison to traditional statistical models, the ANN approach was used for prediction with greater accuracy [72]. ANN algorithms are the most popular for modeling flood prediction since their first usage in the 1990s [73]. Instead of a catchment's physical characteristics, ANNs derive meaning from historical data. Thus, ANNs are considered as reliable data-driven tools for constructing black-box models of complex and nonlinear relationships of rainfall and flood [74], as well as

river flow and discharge forecasting [75]. Furthermore, a number of surveys (e.g., Reference [76]) suggest ANN as one of the most suitable modeling techniques which provide an acceptable generalization ability and speed compared to most conventional models. References [77,78] provided reviews on ANN applications in flood. ANNs were already successfully used for numerous flood prediction applications, e.g., streamflow forecasting [79], river flow [80,81], rainfall-runoff [82], precipitation-runoff modeling [83], water quality [55], evaporation [56], river stage prediction [84], low-flow estimation [85], river flows [86], and river time series [57]. Despite the advantages of ANNs, there are a number drawbacks associated with using ANNs in flood modeling, e.g., network architecture, data handling, and physical interpretation of the modeled system. A major drawback when using ANNs is the relatively low accuracy, the urge to iterate parameter tuning, and the slow response to gradient-based learning processes [87]. Further drawbacks associated with ANNs include precipitation prediction [88,89] and peak-value prediction [90].

The feed-forward neural network (FFNN) [25] is a class of ANN, whereby the network's connections are not in cyclical form. FFNNs are the simplest type of ANN, whereby information moves in a forward direction from input nodes to the hidden layer and later to output nodes. On the other hand, a recurrent neural network (RNN) [91] is a class of ANN, whereby the network's connections form a time sequence for dynamic temporal behavior. Furthermore, RNNs benefit from extra memory to analyze input sequences. In ANNs, backpropagation (BP) is a multi-layered NN where weights are calculated using the propagation of the backward error gradient. In BP, there are more phases in the learning cycle, using a function for activation to send signals to the other nodes. Among various ANNs, the backpropagation ANN (BPNN) was identified as the most powerful prediction tool suitable for flood time-series prediction [26]. Extreme learning machine (ELM) [92] is an easy-to-use form of FFNN, with a single hidden layer. Here, ELM was studied under the scope of ANN methods. ELM for flood prediction recently became of interest for hydrologists and was used to model short-term streamflow with promising results [93,94].

### 3.2. Multilayer Perceptron (MLP)

The vast majority of ANN models for flood prediction are often trained with a BPNN [95]. While BPNNs are today widely used in this realm, the MLP—an advanced representation of ANNs—recently gained popularity [96]. The MLP [97] is a class of FFNN which utilizes the supervised learning of BP for training the network of interconnected nodes of multiple layers. Simplicity, nonlinear activation, and a high number of layers are characteristics of the MLP. Due to these characteristics, the model was widely used in flood prediction and other complex hydrogeological models [98]. In an assessment of ANN classes used in flood modeling, MLP models were reported to be more efficient with better generalization ability. Nevertheless, the MLP is generally found to be more difficult to optimize [99]. Back-percolation learning algorithms are used to individually calculate the propagation error in hidden network nodes for a more advanced modeling approach.

Here, it is worth mentioning that the MLP, more than any other variation of ANNs (e.g., FFNN, BPNN, and FNN), gained popularity among hydrologists. Furthermore, due to the vast number of case studies using the standard form of MLP, it diverged from regular ANNs. In addition, the authors of articles in the realm of flood prediction using the MLP refer to their models as MLP models. From this perspective, we decided to devote a separate section to the MLP.

### 3.3. Adaptive Neuro-Fuzzy Inference System (ANFIS)

The fuzzy logic of Zadeh [100] is a qualitative modeling scheme with a soft computing technique using natural language. Fuzzy logic is a simplified mathematical model, which works on incorporating expert knowledge into a fuzzy inference system (FIS). An FIS further mimics human learning through an approximation function with less complexity, which provides great potential for nonlinear modeling of extreme hydrological events [101,102], particularly floods [103]. For instance, Reference [104] studied river level forecasting using an FIS, as did Lohani et al. (2011) [4] for rainfall-runoff modeling for

water level. As an advanced form of fuzzy-rule-based modeling, neuro-fuzzy presents a hybrid of the BPNN and the widely used least-square error method [46]. The Takagi–Sugeno (T–S) fuzzy modeling technique [4], which is created using neuro-fuzzy clustering, is also widely applied in RFFA [28].

Adaptive neuro-FIS, or so-called ANFIS, is a more advanced form of neuro-fuzzy based on the T–S FIS, first coined [67,77]. Today, ANFIS is known to be one of the most reliable estimators for complex systems. ANFIS technology, through combining ANN and fuzzy logic, provides higher capability for learning [101]. This hybrid ML method corresponds to a set of advanced fuzzy rules suitable for modeling flood nonlinear functions. An ANFIS works by applying neural learning rules for identifying and tuning the parameters and structure of an FIS. Through ANN training, the ANFIS aims at catching the missing fuzzy rules using the dataset [67]. Due to fast and easy implementation, accurate learning, and strong generalization abilities, ANFIS became very popular in flood modeling. The study of Lafdani et al. [60] further described its capability in modeling short-term rainfall forecasts with high accuracy, using various types of streamflow, rainfall, and precipitation data. Furthermore, the results of Shu and [67] showed easier implementation and better generalization capability, using the one-pass subtractive clustering algorithm, which led several rounds of random selection being avoided.

### 3.4. Wavelet Neural Network (WNN)

Wavelet transform (WT) [46] is a mathematical tool which can be used to extract information from various data sources by analyzing local variations in time series [50]. In fact, WT has significantly positive effects on modeling performance [105]. Wavelet transforms supports the reliable decomposition of an original time series to improve data quality. The accuracy of prediction is improved through discrete WT (DWT), which decomposes the original data into bands, leading to an improvement of flood prediction lead times [106]. DWT decomposes the initial data set into individual resolution levels for extracting better-quality data for model building. DWTs, due to their beneficial characteristics, are widely used in flood time-series prediction. In flood modeling, DWTs were widely applied in, e.g., rainfall-runoff [51], daily streamflow [106], and reservoir inflow [107]. Furthermore, hybrid models of DWTs, e.g., wavelet-based neural networks (WNNs) [108], which combine WT and FFNNs, and wavelet-based regression models [109], which integrate WT and multiple linear regression (MLR), were used in time-series predictions of floods [110]. The application of WNN for flood prediction was reviewed in Reference [70], where it was concluded that WNNs can highly enhance model accuracy. In fact, most recently, WNNs, due to their potential in enhancing time-series data, gained popularity in flood modeling [50], for applications such as daily flow [111], rainfall–runoff [112], water level [113], and flash floods [114].

### 3.5. Support Vector Machine (SVM)

Hearst et al. [115] proposed and classified the support vector (SV) as a nonlinear search algorithm using statistical learning theory. Later, the SVM [116] was introduced as a class of SV, used to minimize over-fitting and reduce the expected error of learning machines. SVM is greatly popular in flood modeling; it is a supervised learning machine which works based on the statistical learning theory and the structural risk minimization rule. The training algorithm of SVM builds models that assign new non-probabilistic binary linear classifiers, which minimize the empirical classification error and maximize the geometric margin via inverse problem solving. SVM is used to predict a quantity forward in time based on training from past data. Over the past two decades, the SVM was also extended as a regression tool, known as support vector regression (SVR) [117].

SVMs are today known as robust and efficient ML algorithms for flood prediction [118]. SVM and SVR emerged as alternative ML methods to ANNs, with high popularity among hydrologists for flood prediction. They use the statistical learning theory of structural risk minimization (SRM), which provides a unique architecture for delivering great generalization and superior efficiency. Most importantly, SVMs are both suitable for linear and nonlinear classification, and the efficient mapping of inputs into feature spaces [119]. Thus, they were applied in numerous flood prediction

cases with promising results, excellent generalization ability, and better performance, compared to ANNs and MLRs, e.g., extreme rainfall [120], precipitation [43], rainfall-runoff [121], reservoir inflow [122], streamflow [123], flood quantiles [48], flood time series [124], and soil moisture [125]. Unlike ANNs, SVMs are more suitable for nonlinear regression problems, to identify the global optimal solution in flood models [126]. Although the high computation cost of using SVMs and their unrealistic outputs might be demanding, due to their heuristic and semi-black-box nature, the least-square support vector machine (LS-SVM) highly improved performance with acceptable computational efficiency [127]. The alternative approach of LS-SVM involves solving a set of linear tasks instead of complex quadratic problems [128]. Nevertheless, there are still a number of drawbacks that exist, especially in the application of seasonal flow prediction using LS-SVM [129].

### 3.6. Decision Tree (DT)

The ML method of DT is one of the contributors in predictive modeling with a wide application in flood simulation. DT uses a tree of decisions from branches to the target values of leaves. In classification trees (CT), the final variables in a DT contain a discrete set of values where leaves represent class labels and branches represent conjunctions of features labels. When the target variable in a DT has continuous values and an ensemble of trees is involved, it is called a regression tree (RT) [130]. Regression and classification trees share some similarities and differences. As DTs are classified as fast algorithms, they became very popular in ensemble forms to model and predict floods [131]. The classification and regression tree (CART) [132,133], which is a popular type of DT used in ML, was successfully applied to flood modeling; however, its applicability to flood prediction is yet to be fully investigated [134]. The random forests (RF) method [69,135] is another popular DT method for flood prediction [136]. RF includes a number of tree predictors. Each individual tree creates a set of response predictor values associated with a set of independent values. Furthermore, an ensemble of these trees selects the best choice of classes [69]. Reference [137] introduced RF as an effective alternative to SVM, which often delivers higher performance in flood prediction modeling. Later, Bui et al. [138] compared the performances of ANN, SVM, and RF in general applications to floods, whereby RF delivered the best performance. Another major DT is the M5 decision-tree algorithm [139]. M5 constructs a DT by splitting the decision space and single attributes, thereby decreasing the variance of the final variable. Further DT algorithms popular in flood prediction include reduced-error pruning trees (REPTs), Naïve Bayes trees (NBTs), chi-squared automatic interaction detectors (CHAIDs), logistic model trees (LMTs), alternating decision trees (ADTs), and exhaustive CHAIDs (E-CHAIDs).

### 3.7. Ensemble Prediction Systems (EPSs)

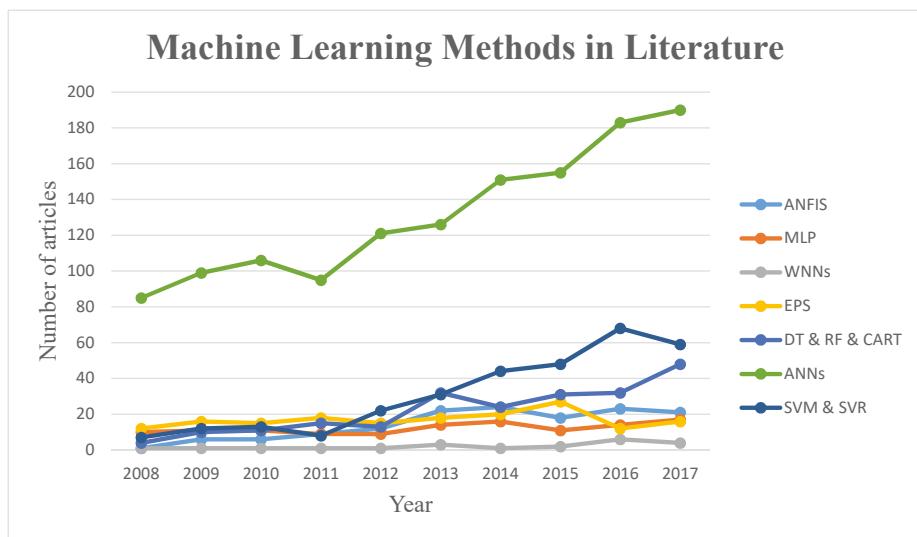
A multitude of ML modeling options were introduced for flood modeling with a strong background [140]. Thus, there is an emerging strategy to shift from a single model of prediction to an ensemble of models suitable for a specific application, cost, and dataset. ML ensembles consist of a finite set of alternative models, which typically allow more flexibility than the alternatives. Ensemble ML methods have a long tradition in flood prediction. In recent years, ensemble prediction systems (EPSs) [141] were proposed as efficient prediction systems to provide an ensemble of  $N$  forecasts. In EPS,  $N$  is the number of independent realizations of a model probability distribution. EPS models generally use multiple ML algorithms to provide higher performance using an automated assessment and weighting system [140]. Such a weighting procedure is carried out to accelerate the performance evaluation process. The advantage of EPS is the timely and automated management and performance evaluation of the ensemble algorithms. Therefore, the performance of EPS, for flood modeling in particular, can be improved. EPSs may use multiple fast-learning or statistical algorithms as classifier ensembles, e.g., ANNs, MLP, DTs, rotation forest (RF) bootstrap, and boosting, allowing higher accuracy and robustness. The subsequent ensemble prediction systems can be used to quantify the probability of floods, based on the prediction rate used in the event [142–144]. Therefore, the quality of ML ensembles can be calculated based on the verification of probability distribution.

Ouyang et al. [145] and Zhang et al. [146] presented a review of the applications of ensemble ML methods used for floods. EPSSs were demonstrated to have the capability for improving model accuracy in flood modeling [140–146].

To improve the accuracy of import data and to achieve better dataset management, the ensemble mean was proposed as a powerful approach coupled with ML methods [140,141]. Empirical mode decomposition (EMD) [142], and ensemble EMD (EEMD) [143] are widely used for flood prediction [144]. Nevertheless, EMD-based forecast models are also subject to a number of drawbacks [145]. The literature includes numerous studies on improving the performance of decomposition and prediction models in terms of additivity and generalization ability [146].

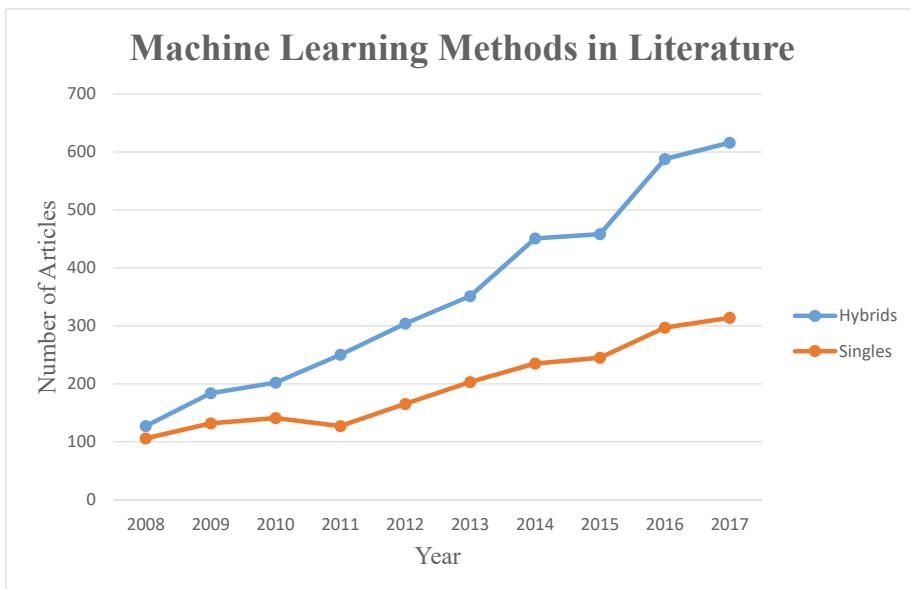
### 3.8. Classification of ML Methods and Applications

The most popular ML modeling methods for flood prediction were identified in the previous section, including ANFIS, MLP, WNN, EPS, DT, RF, CART, and ANN. Figure 3 presents the major ML methods used for flood prediction, and the number of corresponding articles in the literature over the last decade. This figure was designed to communicate to the readers which ML methods increased in popularity among hydrologists for flood modeling within the past decade.



**Figure 3.** Major ML methods used for flood prediction in the literature. Reference year: 2008 (source: Scopus).

Considering the ML methods for application to floods, it is apparent that ANNs, SVMs, MLPs, DTs, ANFIS, WNNs, and EPSSs are the most popular. These ML methods can be categorized as single and hybrid methods. In addition to the fundamental hybrid ML methods, i.e., ANFIS, WNNs, and basic EPSSs, several different research strategies for obtaining better prediction evolved [137]. The strategies involved developing hybrid ML models using soft computing techniques, statistical methods, and physical models rather than individual ML approaches, whereby the extra components complement each other with respect to their drawbacks and shortcomings. The success of such hybrid approaches motivated the research community to explore more advanced hybrid models. Figure 4 presents the progress of single vs. hybrid ML methods for flood prediction in the literature over the past decade. The figure shows an apparent continuous increase and notable progress in using novel hybrid methods. Through Figure 4, the taxonomy of our research was justified, based on distinguishing hybrid and single ML prediction models.



**Figure 4.** The progress of single vs. hybrid ML methods for flood prediction in the literature. Reference year: 2008 (source: Scopus).

Furthermore, the types of prediction are often studied with different lead-time predictions due to the flood. Real-time, hourly, daily, weekly, monthly, seasonal, annual, short-term, and long-term are the terms most often used in the literature. Real-time prediction is concerned with anywhere between few minutes and an hour preceding the flood. Hourly predictions can be 1–3 h ahead of the flood forecasting lead time or, in some cases, 18 h or 24 h. Daily predictions can be 1–6 days ahead of the forecast. Monthly forecasts can be, for instance, up to three months. In hydrology, the definitions of short-term and long-term in studying the different phenomena vary. Short-term predictions for floods often refer to hourly, daily, and weekly predictions, and they are used as warning systems. On the other hand, long-term predictions are mostly used for policy analysis purposes. Furthermore, if the prediction leading time to flood is three days longer than the confluence time, the prediction is considered to be long-term [37,58]. From this perspective, in this study, we considered a lead time greater than a week as a long-term prediction. It was observed that the characteristics of the ML methods used varied significantly according to the period of prediction. Thus, dividing the survey on the basis of short-term and long-term was essential.

Here, it is also worth emphasizing that, in this paper, the prediction lead-time was classified as “short-term” or “long-term”. Although flash floods happen in a short period of time with great destructive power, they can be predicted with either “short-term” or “long-term” lead times to the actual flood. In fact, this paper is concerned with the lead times instead of the duration or type of flood. If the lead-time prediction to a flash flood was short-term, then it was studied as a short-term lead time. However, sometimes flash floods can be predicted with long lead times. In other words, flash floods might be predicted one month ahead. In this case, the prediction was considered as long-term. Regardless of the type of flood, we only focused on the lead time.

In this study, the ML methods were reviewed using two classes—single methods and hybrid methods. Figures 5 and 6 represent the taxonomy of the research.

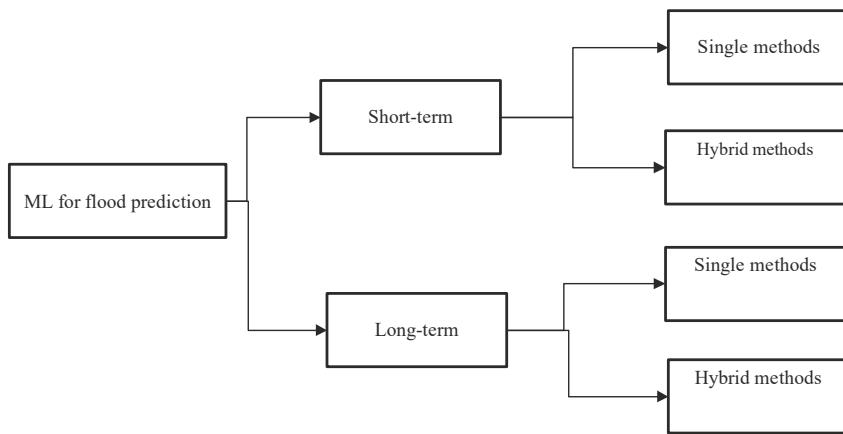


Figure 5. Taxonomy of the survey—ML methods for flood prediction.

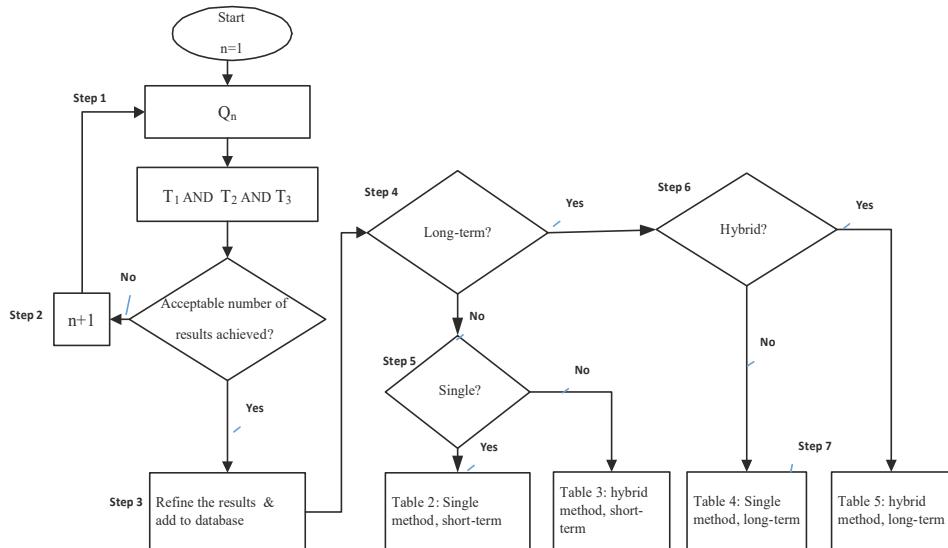


Figure 6. Taxonomy of the survey.

Step 1 involved running the queries one by one; step 2 involved checking the results of the search, and initiating the next search; step 3 involved identifying the comparative studies on ML models of prediction, refining the results and building the database; step 4 involved identifying whether it was a long-term or short-term prediction; steps 5 and 6 involved identifying if it was a single or hybrid method, constructing Table 1, and step 7 involved constructing the other Tables. The four tables provide the list of studies on different prediction techniques, which entail the organized comprehensive surveys of the literature.

#### 4. Short-Term Flood Prediction with ML

Short-term lead-time flood predictions are considered important research challenges, particularly in highly urbanized areas, for timely warnings to residences so to reduce damage [146]. In addition,

short-term predictions contribute highly to water recourse management. Even with the recent improvements in numerical weather prediction (NWP) models, artificial intelligence (AI) methods, and ML, short-term prediction remains a challenging task [147–152]. This section is divided into two subsections—single and hybrid methods of ML—to individually investigate each group of methods.

#### 4.1. Short-Term Flood Prediction Using Single ML Methods

To gain insight into the performance of ML methods, a comprehensive comparison was required to investigate ML methods. Table 1 presents a summary of the major ML methods, i.e., ANNs, MLP, nonlinear autoregressive network with exogenous inputs (NARX), M5 model trees, DTs, CART, SVR, and RF, followed by a comprehensive performance comparison of single ML methods in short-term flood prediction. A revision and discussion of these methods follow so as to identify the most suitable methods presented in the literature.

**Table 1.** Short-term predictions using single machine learning (ML) methods.

Modeling Technique	Reference	Flood Resource Variable	Prediction Type	Region
ANN vs. statistical	[1]	Streamflow and flash flood	Hourly	USA
ANN vs. traditional	[44]	Water and surge level	Hourly	Japan
ANN vs. statistical	[149]	Flood	Real-time	UK
ANN vs. statistical	[150]	Extreme flow	Hourly	Greece
FFANN vs. ANN	[151]	Water level	Hourly	India
ANN vs. T-S	[4]	Flood	Hourly	India
ANN vs. AR	[153]	Stage level and streamflow	Hourly	Brazil
MLP vs. Kohonen NN	[154]	Flood frequency analysis	Long-term	China
BPANN	[155]	Peak flow of flood	Daily	Canada
BPANN vs. DBPANN	[156]	Rainfall-runoff	Monthly and daily	China
BPANN	[157]	Flash flood	Real-time	Hawaii
BPANN	[158]	Runoff	Daily	India
ELM vs. SVM	[159]	Streamflow	Daily	China
BPANN vs. NARX	[160,161]	Urban flood	Real-time	Taiwan
FFANN vs. Functional ANN	[162]	River flows	Real-time	Ireland
Recurrent NN vs. Z-R relation	[163]	Rainfall prediction	Real-time	Taiwan
ANN vs. M5 model tree	[164]	Peak flow	Hourly	India
NBT vs. DT vs. Multinomial regression	[165]	Flash flood	Real-time, hourly	Austria
DTs vs. NBT vs. ADT vs. LMT, and REPT	[166]	Flood	Hourly/daily	Iran
MLP vs. MLR	[167,168]	River flow and rainfall-runoff	Daily	Algeria
MLP vs. MLR	[98]	River runoff	Hourly	Morocco
MLP vs. WT vs. MLR vs. ANN	[169]	River flood forecasting	Daily	Canada
ANN vs. MLP	[170]	River level	Hourly	Ireland
MLP vs. DT vs. CART vs. CHAID	[171]	Flood during typhoon	Rainfall-runoff	China
SVM vs. ANN	[120]	Rainfall extreme events	Daily	India
ANN vs. SVR	[48]	Flood	Daily	Canada
RF vs. SVM	[69]	Rainfall	Hourly	Taiwan

Kim and Barros [148] modified an ANN model to improve flood forecasting short-term lead time through consideration of atmospheric conditions. They used satellite data from the ISCCP-B3 dataset [172]. This dataset includes hourly rainfall from 160 rain gauges within the region. The ANN was reported to be considerably more accurate than the statistical models. In another similar work, Reference [44] developed an ANN forecast model for hourly lead time. In their study, various datasets were used, consisting of meteorological and hydrodynamic parameters of three typhoons. Testing of the ANN forecast models showed promising results for 5-h lead time. In another attempt, Danso-Amoako [1] provided a rapid system for predicting floods with an ANN. They provided a reliable forecasting tool for rapidly assessing floods. An  $R^2$  value of 0.70 for the ANN model proved that the tool was suitable for predicting flood variables with a high generalization ability. The results of [149] provides similar conclusions. Furthermore, Panda, Pramanik, and Bala [151] compared the accuracy of ANN with FFANN, and the results were benchmarked with the physical model of MIKE 11 for short-term water level prediction. This dataset includes the hourly discharge and water level between 2006 and 2009. The data of the year 2006 was used for testing root-mean-square error (RMSE). The results indicated that the FFANN performed faster and relatively more accurately than the ANN model. Here, it is worth mentioning that the overall results indicated that the neural networks were superior compared to the one-dimensional model MIKE 11. Nevertheless, there were great advancements reported in the implementation of two-dimensional MIKE 11 [8].

Kourgialas, Dokou, and Karatzas [150] created a modeling system for the prediction of extreme flow based on ANNs 3 h, 12 h, and 19 h ahead of the flood. They analyzed five years of hourly data to investigate the ANN effectiveness in modeling extreme flood events. The results indicated it to be highly effective compared to conventional hydrological models. Lohani, Goel, and Bhatia [4] improved the real-time forecasting of rainfall-runoff of floods, and the results were compared to the T-S fuzzy model and the subtractive-clustering-based T-S (TSC-T-S) fuzzy model. They, however, concluded that the fuzzy model provided more accurate predictions with longer lead time. The hourly rainfall data from 1989 to 1995 of a gauge site, in addition to the rainfall during a monsoon, was used. Pereira Filho and dos Santos [153] compared the AR model with an ANN in simulating forecast stage level and streamflow. The dataset was created from independent flood events, radar-derived rainfall, and streamflow rain gauges available between 1991 and 1995. The AR and ANN were employed to model short-term flood in an urban area utilizing streamflow and weather data. They showed that the ANN performed better in its verification and it was proposed as a better alternative to the AR model.

Ahmad and Simonovic [155] used a BPNN for predicting peak flow utilizing causal meteorological parameters. This dataset included daily discharge data for 1958–1997 from gauging stations. BPNN proved to be a fast and accurate approach with the ability of generalization for application to other locations with similar rivers. Furthermore, to improve the simulation of daily streamflow using BPNN, Reference [156] used division-based backpropagation to obtain satisfying results. The raw data of local evaporation and rainfall gauges of six years were used for the short-term flood prediction of a streamflow time series. The dataset of one decade from 1988 was used for training and the dataset of five subsequent years was used for testing. The BPNN model provided promising results; however, it lacked efficiency in using raw data for the time-series prediction of streamflow. In addition, Reference [157] showed the application of BPNN for assessing flash floods using measured data. This dataset included 5-min-frequency water quality data and 15-min-frequency rainfall data of 20 years from two rain gauge stations. Their experiments introduced ANN models as simple ML methods to apply, while simultaneously requiring expert knowledge by the user. In addition, their ANN prediction model showed great ability to deal with a noisy dataset. Ghose [158] predicted the daily runoff using a BPNN prediction model. The data of daily water level of two years from 2013–2015 were used. The accurate BPNN model was reported with an efficiency of 96.4% and an  $R^2$  of 0.94 for flood prediction.

Pan, Cheng, and Cai [159] compared the performances of ELM and SVM for short-term streamflow prediction. Both methods demonstrated a similar level of accuracy. However, ELM was suggested

as a faster method for parameter selection and learning loops. Reference [154] also conducted a comparison between fuzzy c-means, ANN, and MLP using a common dataset of sites to investigate ML method efficiency and accuracy. The MLP and ANN methods were proposed as the best methods. Chang, Chen, Lu, Huang, and Chang [160] and Reference [161] modeled multi-step urban flood forecasts using BPNN and a nonlinear autoregressive network with exogenous input (NARX) for hourly forecasts. The results demonstrated that NARX worked better in short-term lead-time prediction compared to BPNN. The NARX network produced an average  $R^2$  value of 0.7. This study suggested that the NARX model was effective in urban flood prediction. Furthermore, Valipour et al. [24] showed how the accuracy of ANN models could be increased through integration with autoregressive (AR) models.

Bruen and Yang [162] modeled real-time rainfall–runoff forecasting for different lead times using FFNN, ARMA, and functional networks. Here, functional networks [173] were compared with an FFNN model. The models were tested using a storm time-series dataset. The result was that functional networks allowed quicker training in the prediction of rainfall–runoff processes with different lead times. The models were able to predict floods with short lead times. Reference [164] estimated water level–discharge using M5 trees and ANN. This dataset was collected from the period of 1990 to 1998, and the inputs were supplied by computing the average mutual information. The ANN and M5 model tree performed similar in terms of accuracy. Reference [166] tested four DT models, i.e., alternating decision trees (ADTs), reduced-error pruning trees (REPTs), logistic model trees (LMTs), and NBTs, using a dataset of 200 floods. The ADT model was reported to perform better for flash-flood prediction for a speedy determination of flood-susceptible areas. In other research, Reference [165] compared the performance of an NBT and DT prediction model, using geomorphological disposition parameters. Both models and their hybrids were compared in terms of prediction accuracy in a catchment. The advanced DTs were found to be promising for flood assessment in prone areas. They concluded that an independent dataset and benchmarking of other ML methods were required for judgment of the accuracy and efficiency of the method. Reference [171] worked on a dataset including more than 100 tropical cyclones (TCs) affecting a watershed for the hourly prediction of precipitation. The performances of MLP, CART, CHAID, exhaustive CHAID, MLR, and CLIM were compared. The evaluation results showed that MLP and DTs provided better prediction. Reference [163] applied a dynamic ANN, as well as a Z–R relation approach for constructing a one-hour-ahead prediction model. This dataset included three-dimensional radar data of typhoon events and rain gauges from 1990 to 2004, including various typhoons. The results indicated that the ANN performed better.

Aichouri, Hani, Bougerira, Djabri, Chaffai, and Lallahem [167] implemented an MLP model for flood prediction, and compared the results with the traditional MLR model. The rainfall–runoff daily data from 1986 to 2003 were used for model building. The results and comparative study indicated that the MLP approach performed with better yield for river rainfall–runoff. In a similar research, Reference [98] modeled and predicted the river rainfall–runoff relationship through training six years of collected daily rainfall data using MLP and MLR (1990 to 1995). Furthermore, the data of 1996 were used for testing to select the best performing network model. The  $R^2$  values for the ANN and MLR models were 0.888 and 0.917, respectively, showing that the MLP approach gave a much better prediction than MLR. Reference [169] proposed a number of data-based flood predictions for daily stream flows models using MLP, WT, MLR, ARIMA, and ANN. This dataset included two time series of streamflow and a meteorological dataset including records from 1970 to 2001. The results showed that MLP, WT, and ANN performed generally better. However, the proposed WT prediction model was evaluated to be not as accurate as ANN and MLP for a one-week lead time. Reference [170] designed optimal models of ANN and MLP for the prediction of river level. This study indicated that an optimization tool for the ANN network can highly improve prediction quality. The candidate inputs included river levels and mean sea-level pressure (SLP) for the period of 2001–2002. The MLP was identified as the most accurate model for short-term river flood prediction.

Nayak and Ghosh [120] used SVM and ANN to predict hourly rainfall–runoff using weather patterns. A model of SVM classifier for rainfall prediction was used and the results were compared to ANN and another advanced statistical technique. The SVM model appeared to predict extreme floods better than the ANN. Furthermore, the SVM model proved to function better in terms of uncertainty. Gizaw and Gan [48] developed SVR and ANN models for creating RFFA to estimate regional flood quantiles and to assess climate change impact. This dataset included daily precipitation data obtained from gauges from 1950 to 2016. RMSE and  $R^2$  were used for the evaluation of the models. The SVR model estimated regional flood more accurately than the ANN model. SVR was reported to be a suitable choice for predicting future flood under the uncertainty of climate change scenarios [118]. In a similar attempt, Reference [69] provided effective real-time flood prediction using a rainfall dataset measured by radar. Two models of RF and SVM were developed and their prediction performances were compared. Their performance comparison revealed the effectiveness of SVM in real-time flood forecasting.

Table 2 represents a comparative analysis of single ML models for the prediction of short-term floods, considering the complexity of the algorithm, ease of use, running speed, accuracy, and input dataset. This table was created based on the revisions that were made on the articles of Table 1 and also the accuracy analysis of Figure 3, where the values of  $R^2$  and RMSE of the single ML methods were considered. The quality of ML model prediction, in terms of speed, complexity, accuracy, and ease of use, was continuously improved through using ensembles of ML methods, hybridization of ML methods, optimization algorithms, and/or soft computing techniques. This trend of improvement is discussed in detail in the discussion.

**Table 2.** Comparative analysis of single ML models for the prediction of short-term floods.

Modeling Technique	Complexity of Algorithm	Ease of Use	Speed	Accuracy	Input Dataset
ANN	High	Low	Fair	Fair	Historical
BPANN	Fairly high	Low	Fairly high	Fairly high	Historical
MLP	Fairly high	Fair	High	Fairly high	Historical
ELM	Fair	Fairly high	Fairly high	Fair	Historical
CART	Fair	Fair	Fair	Fairly high	Historical
SVM	Fairly high	Low	Low	Fair	Historical
ANFIS	Fair	Fairly high	Fair	Fairly high	Historical

#### 4.2. Short-Term Flood Prediction Using Hybrid ML Methods

To improve the quality of prediction, in terms of accuracy, generalization, uncertainty, longer lead time, speed, and computation costs, there is an ever increasing trend in building hybrid ML methods. These hybrid methods are numerous, including more popular ones, such as ANFIS and WNN, and further novel algorithms, e.g., SVM–FR, HEC–HMS–ANN, SAS–MP, SOM–R–NARX, wavelet-based NARX, WBANN, WNN–BB, RNN–SVR, RSVRCPSO, MLR–ANN, FFRM–ANN, and EPSs. Table 3 presents these methods; a revision of the methods and applications follows along with a discussion on the ML methods.

**Table 3.** Short-term flood prediction using hybrid ML methods.

Modeling Technique	Reference	Flood resource Variable	Prediction Type	Region
ANFIS vs. ANN	[174]	Flash floods	Real-time	Spain
ANFIS vs. ANN	[175,176]	Water level	Hourly	Taiwan
ANFIS vs. ANN	[46]	Watershed rainfall	Hourly	Taiwan
ANFIS vs. ANN	[67]	Flood quantiles	Real-time	Canada
ANN vs. ANFIS	[177]	Daily flow	Daily	Iran
CART vs. ANFIS vs. MLP vs. SVM	[134]	Sediment transport	Daily	Iran
MLP vs. GRNNM vs. NNM	[96]	Flood prediction	Daily	Korea
SVM-FR vs. DT	[178]	Rainfall-runoff	Real-time	Malaysia
HEC-HMS-ANN vs. HEC-HMS-SVR	[179]	Rainfall-runoff	Hourly	Taiwan
SAS-MP vs. W-SAS-MP	[180]	Flash flood and streamflow	Daily	Turkey
SOM-R-NARX vs. R-NARX	[181]	Regional flood	Hourly	Taiwan
Wavelet-based NARX vs. ANN, vs. WANN	[182]	Streamflow forecasting	Daily	India
WBANN vs. WANN vs. ANN vs. BANN	[105]	Flood	Hourly	India
ANN-hydrodynamic model	[183]	Flood prediction: tidal surge	Hourly	UK
RNN-SVR, RSVRCPSO	[184]	Flash flood: rainfall forecasting	Hourly	Taiwan
AME and SSNN vs. ANN	[185]	Rainfall forecasting	Hourly	Taiwan
Hybrid of FFNN with linear model	[186]	Flood forecasting: daily flows	Daily	India
FFNN vs. FBNN vs. FFRM-ANN	[187]	Flash floods	Hourly	Taiwan
ANN-NLPM vs. ANN	[188]	Rainfall-runoff	Daily	China
EPS of MLP vs. SVM vs. RF	[189]	Runoff simulations	Real-time	Germany
EPS of ANNs	[190]	Flood	Daily	Canada

Jimeno-Sáez, et al. [174] modeled flash floods using ANN and ANFIS, applying a dataset collected from 14 different streamflow gauge stations. RMSE and  $R^2$  were used as evaluation criteria. The results showed that ANFIS demonstrated a considerably superior ability to estimate real-time flash floods compared to ANN. Chang and Chang [175] constructed an accurate water level forecasting system based on ANFIS for 1–3 h ahead of the flood. The ANFIS successfully provided accurate water level prediction. The hourly water level of five gauges from 1971 to 2001 was used. They concluded that the ANFIS model could efficiently deal with a big dataset [176] through fast learning and reliable prediction. A further comparison showed that the ANFIS hybrid model tuned by SVR provided superior prediction accuracy and good cost-effective computation for nonlinear and real-time flood prediction. In addition, the model with human interaction could provide better performance. In another similar research, Reference [46] developed an ANFIS model based on a precipitation dataset, which provided reliable hourly predictions with an  $R^2$  more than 0.85. The results were reported as highly satisfactory for the typhoon season. Reference [67] used ANFIS for ungauged sites of 151 catchments; the results were evaluated and compared to the ANN, NLR, NLR-R modes using a Jackknife procedure. The evaluation showed that the ANFIS model provided higher generalization capability compared to the NLR and ANN models. The ANFIS model implemented an efficient mechanism for forecasting the flood region, and providing insight from the data, leading to prediction. Rezaeianzadeh (2014) [177] presented a number of forecasting systems for daily flow prediction using ANN, ANFIS, MLR, and MNLR. Furthermore, the performances of the models were calculated with RMSE and  $R^2$ . This dataset

included precipitation data from various meteorological stations. Furthermore, the evaluation showed that MNLR models with lower RMSE values had a better performance than the ANFIS, MLR, and ANN models. Furthermore, MNLR was suggested as a low-cost and efficient model for the daily prediction of flow. In a similar attempt, Choubin, Darabi et al. (2018) [133] evaluated the accuracy of ANFIS, considering three common ML modeling tools—CART, SVM, and MLP. The evaluation suggested that the CART model performed best. Therefore, CART was strongly suggested as a reliable prediction tool for hydro-meteorological datasets. Kim and Singh [96] developed three models, namely generalized regression ANN (GRNNM), Kohonen self-organizing feature maps ANN (KSOFM–NNM), and MLP, for flood prediction. Furthermore, the prediction performance was evaluated, showing that KSOFM–NNM performed accurately compared to MLP and GRNNM in forecasting flood discharge. The hybrid models, overall, were shown to overcome the difficulties when using single ANN models. Reference [178] proposed an advanced ensemble model through combining FR and SVM to build spatial modeling in flood prediction. The results were compared with DT. This dataset included an inventory map of flood prediction in various locations. To build the model, up to 100 flood locations were used for training and validation. The evaluation results showed a high success rate for the ensemble model. The results proved the efficiency, accuracy, and speed of the model in the susceptibility assessment of floods.

Young, Liu, and Wu [179] developed a hybrid physical model through integrating the HEC–HMS model with SVM and ANN for accurate rainfall–runoff modeling during a typhoon. The hybrid models of HEC–HMS–SVR and HEC–HMS–ANN had acceptable capability for hourly prediction. However, the SVR model had much better generalization and accuracy ability in runoff discharge predictions. It was concluded that the predictions of HEC–HMS were improved through ML hybridization. Reference [180] proposed SAS–MP, which is a hybrid of wavelet and season multilayer perceptron for daily rainfall prediction. The season algorithm is a novel decomposition technique used to improve data quality. The resulting hybrid model was referred to as the W-SAS–MP model. This dataset included the daily rainfall data of three decades since 1974. The W-SAS–MP model was reported as highly efficient for enhancing daily rainfall prediction accuracy and lead time.

Chang, Shen, and Chang [181] developed a hybrid ANN model for real-time forecasting of regional floods in an urban area. The advanced hybrid model of SOM–R-NARX was an integration of the NARX network with SOM. Their big dataset included 55 rainfall events of daily rainfall. The evaluation suggested that SOM–R-NARX was accurate with small values of RMSE and high  $R^2$ . Furthermore, compared to the cluster-based hybrid inundation model (CHIM), it provided hourly prediction accuracy. Reference [182] proposed a model of wavelet-based NARX (WNARX) for the daily forecasting of rainfalls on a dataset of gauge-based rainfall data for the period from 2000 to 2010. The prediction performance was further benchmarked with ANN, WANN, ARMAX, and NARX models, whereby WNARX was reported as superior.

Partial [110] developed a model for the daily prediction of precipitation with ANN and WNN models. In their case, WNN showed significantly better results with an average value of 0.79 at various stations. In Reference [60], they compared WNN with ANFIS for daily rainfall. The results showed that the hybrid algorithm of WNN performed better with an  $R^2$  equal to 0.9 for daily lead time. Reference [105] proposed a hybrid model of wavelet, bootstrap technique, and ANN, which they called WBANN. It improved the accuracy and reliability of the ANN model short-term flood prediction. The performance of WBANN was compared with bootstrap-based ANNs (BANNs) and WNN. The wavelet decomposition significantly improved the ANN models. In addition, the bootstrap resampling produced consistent results. French, Mawdsley, Fujiyama, and Achuthan [183] proposed a novel hybrid model of ANN and a hydrodynamic model for the accurate short-term prediction of extreme storm surge water. The ANN–hydrodynamic model generated realistic flood extents and a great improvement in model accuracy. Reference [184] proposed a hybrid forecasting technique called RSVRCPSO to accurately estimate the rainfall. RSVRCPSO is an integration of RNN, SVR, and a chaotic particle swarm optimization algorithm (CPSO). This dataset was obtained from three

rain gauges from the period of 1985 to August 1997, which included the data of nine typhoon events. The results suggested that the proposed model yielded better performance for rainfall prediction. The RSVRCPSO model, in comparison with SVRCPSO, resulted in less RMSE learning and testing, which gave way to superiority in prediction.

Pan et al. [185] proposed a monsoon rainfall enhancement (AME) based on ANNs, which was a hybrid form of linear regression and a state-space neural network (SSNN). The performance of the proposed model was benchmarked against the hybrid method of MLR-ANN. This dataset included the total rain, wind, and humidity measures from 1989–2008 based on 371 rain gauge stations of six typhoons. The results indicated that the method was highly robust with a better prediction accuracy in terms of  $R^2$ , peak discharge, and total volume. Rajurkar et al. [186] modeled rainfall-runoff by integrating ANN and a simplified linear model. Furthermore, this dataset included the daily measurements of rainfall in the period of 1963–1990. The hybrid model was found to be better for providing a theoretical forecasting representation of floods with  $R^2$  equal to 0.728.

Hsu et al. [187] proposed a hybrid model from the integration of a flash-flood routing model (FFRM) and ANN, called the FFRM-ANN model, to predict hourly river stages. The ANN algorithms used in this study were the FFNN and FBNN. Data from eight typhoon events between 2004 and 2005 of rainfall and river stage pairs were selected for model training. The results indicated that the hybrid model of FFRM-ANN provided an efficient FFRM for accurate flood forecasting. The comparison of the hybrid method against each algorithm used in the study proved the effectiveness of the proposed method. Reference [188] developed a hybrid prediction model by integrating ANN and a nonlinear perturbation model (NLPM), defined as NLPM-ANN, to improve the efficiency and accuracy of rainfall-runoff prediction. The model of NLPM-ANN was benchmarked against two models of nonlinear perturbation model (LPM), and NLPM integrated with antecedent precipitation index (API) i.e., NLPM-API, on a dataset of daily rainfall-runoff in the period of 1973–1999. They reported that the NLPM-ANN worked better than the models of LPM and NLPM-API. The results of the case studies of various watersheds proved the model accuracy.

Through an EPS model, Reference [189] aimed at limiting the range of the uncertainties in runoff simulations and flood prediction. The classifier ensembles included MLP, SVM, and RF. Note that the ensemble of MLP was a novel approach in flood prediction. The proposed EPS presented a number of integrated models and simulation runs. The model validation was successfully performed using a dataset from various rain gauges of precipitation data during the 2013–2014 storm season. Using the EPS model decreased uncertainty in forecasting, which resulted in the prediction system being evaluated as reliable and robust in estimating flood duration and destructive power. In another case, Reference [190] developed an EPS model of six ANNs for daily streamflow prediction based on daily high-flow data from the storm season of 2013–2014. The proposed model had a fast development time, which also provided probabilistic forecasts to deal with uncertainties in prediction. The ensemble prediction system was reported as highly useful and robust.

#### 4.3. Comparative Performance Analysis

To evaluate a reliable prediction, the accuracy, reliability, robustness, consistency, generalization, and timeliness are suggested as the basic criteria (Singh 1989). The timeliness is one of the most important criteria, and it is only achieved through using robust yet simple models. Furthermore, the performance of the prediction models is often evaluated through root-mean-square error (RMSE), mean error (ME), mean squared error (MSE), Nash coefficients (E), and  $R^2$ , also known as the correlation coefficient (CC). In this survey, the values of  $R^2$  and RMSE were considered for performance evaluation. CC (Equation (1)) and RMSE (Equation (2)) can be defined as follow:

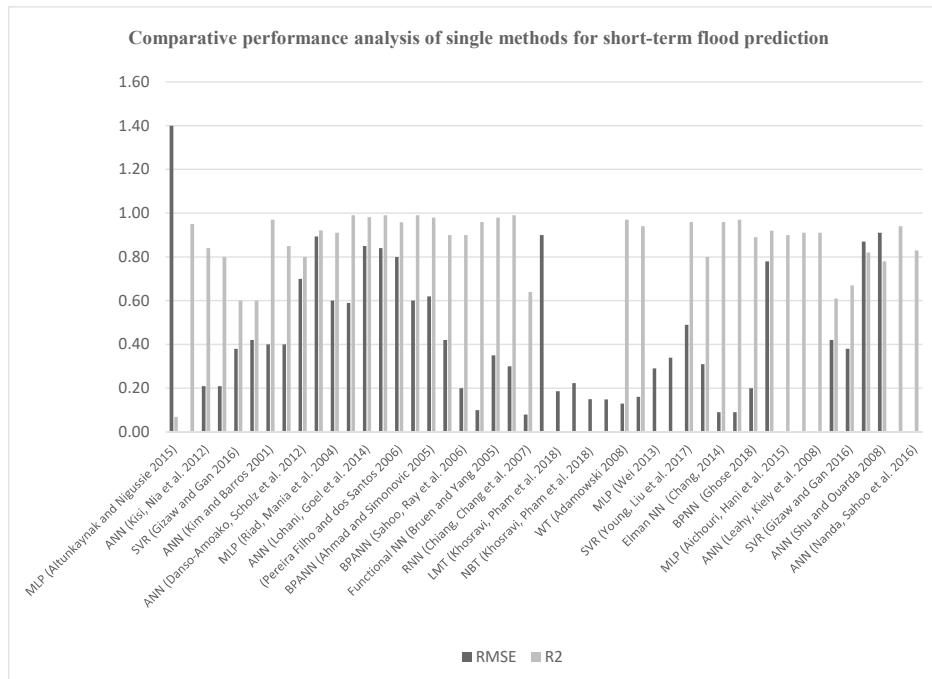
$$CC = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^N (x_i - \bar{x})^2][\sum_{i=1}^N (y_i - \bar{y})^2]}} \quad (1)$$

where  $x_i$  and  $y_i$  are the observed and predicted values and the  $i$ -th residue;  $\bar{x}$  and  $\bar{y}$  are their means, respectively.

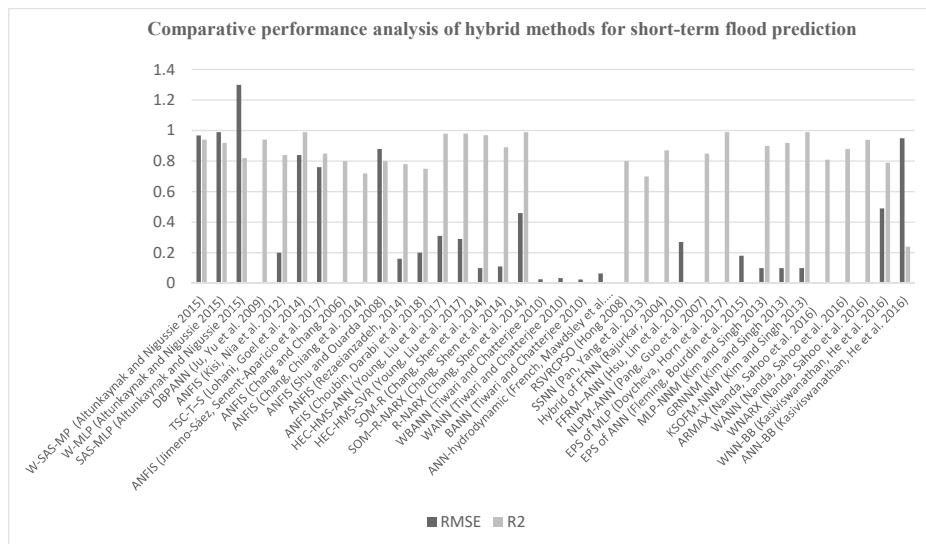
$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{n}} \quad (2)$$

where  $X_{obs}$  defines observed variables and  $X_{model}$  prediction values for year  $i$ , where generally  $R^2 > 0.8$  is considered as an acceptable prediction. However, a lower value for RMSE suggests a better prediction. Overall, forecasting models of floods are reported as accurate if RMSE values are close to 0, and  $R^2$  values are close to 1. The specific intended purpose, computational cost, and dataset would be our major consideration criteria. Furthermore, the generalization ability, speed and cost of implementation and operation, ease of use, low-cost maintenance, robustness, and accuracy of the simulation are other important criteria for evaluation of the methods.

Here, it is worth mentioning that the value of RMSE can be different across various studies. In addition, the values of RMSE in some studies were calculated for various sites. To present a fair evaluation of RMSE, we made sure that the unit of RMSE was the same, and, for the multiple RMSEs, the average was calculated. We also double-checked for any possible error. The comparative performance analysis of single and hybrid ML methods for short-term flood prediction using  $R^2$  and RMSE are presented in Figures 7 and 8 respectively.



**Figure 7.** Comparative performance analysis of single methods of ML for short-term flood prediction using  $R^2$  and root-mean-square error (RMSE).



**Figure 8.** Comparative performance analysis of hybrid methods of ML for short-term flood prediction, using  $R^2$  and RMSE.

Generally, ANNs are suggested as promising means for short-term prediction. Despite performing weakly in a few early studies, especially in the generalization aspect, better methodologies for higher-performance ANNs in handling big datasets yielded better results. In this context, the BPNN and functional networks are suggested as being difficult to be implemented by the user. However, the models were shown to be reasonably accurate, efficient, and fast with the ability to deal with noisy datasets. However, the NARX network performed better compared to BPNN. Nevertheless, accuracy could be enhanced through integration with autoregressive models. MLP and DTs provide equally acceptable prediction yields with ANNs. Among DTs, the ADT model provided the fastest and most accurate prediction capability in determining floods. Although not as popular as ANNs, the rotation forest (RF) and M5 model tree (MT) were reported as efficient and robust. References e.g. [69,136] proposed RF-based models that were as effective as ANNs and suitable for long lead times.

Along with ANNs, the SVM was also seen as a relatively effective ML tool for rainfall-runoff modeling and classification with better generalization ability and performance. In many cases, SVM performed even better, especially for very short lead times [122,125]. In particular, SVM-based models provided promising performances for hourly prediction. Nevertheless, the prediction ability decreased for longer lead times. This issue was addressed using the LS-SVM model, which also showed better generalization ability [127]. Generally, SVM was reported to be a suitable choice to evaluate the uncertainty in predicting hazardous flood quantiles, which revealed the effectiveness of SVM in real-time flood forecasting.

Overall, the reviewed single prediction models could provide relatively accurate short-term forecasts. However, for predictions longer than 2 h, hybrid models such as ANFIS, and WNN performed better. The performance comparisons of the ANFIS model with BPNN and AR models, with average correlation coefficients higher than 0.80, showed the superiority of ANFIS in a wide range of short-term flood prediction applications, e.g., water level, rainfall-runoff, and streamflow (for up to 24 h). ANFIS demonstrated a considerably superior ability for estimating real-time flash flood estimation compared to most ANN-based models, particularly 1–3 h ahead of flood, providing high accuracy and reliability. More advanced ANFIS hybrid models tuned by SVR provided even better prediction accuracy and good cost-effective computation for nonlinear and real-time flood prediction.

Furthermore, ANFIS models presented higher generalization ability. However, by increasing the prediction lead time,  $R^2$  decreased. For daily flow, MNLR was suggested with a superior performance over the ANN, ANFIS, and MLR models. In cases where hydro-meteorological data are readily available, CART was superior to ANFIS, SVM, and MLP; T-S fuzzy was also a good choice. On the other hand, WNN performed significantly better than MLP, ANNs, and ANFIS for daily predictions. For accurate longer lead-time predictions, decomposition techniques such as DWT, autoregression, and the season algorithm provided great advantages.

Overall, the novel hybrid models designed using ML, soft computing, and statistical methods, e.g., KSOFM–NNM, SOM–R–NARX, WNARX, HEC–HMS–SVR, HEC–HMS–ANN, W–SAS–MP, WBANN, RSVRCPSO, and the ANN–hydrodynamic model, were shown to overcome the drawbacks of most ML methods by enhancing the prediction accuracy and lead time, leading to more realistic flood models with even better susceptibility assessment. On the other hand, novel ensemble methods not only improved the accuracy robustness of predictions, but also contributed to limiting the range of uncertainties in models. Among the EPS methods, the ensembles of ANN, MLP, SVM, and RF showed promising results.

## 5. Long-Term Flood Prediction with ML

Long-term flood prediction is of significant importance for increasing knowledge and water resource management potential over longer periods of time, from weekly to monthly and annual predictions [191]. In the past decades, many notable ML methods, such as ANN [74], ANFIS [68,192], SVM [193], SVR [193], WNN [51], and bootstrap–ANN [51], were used for long lead-time predictions with promising results. Recently, in a number of studies (e.g., References [55,194–198]), the performances of various ML methods for long lead-time flood predictions were compared. However, it is still not clear which ML method performs best in long-term flood prediction. In this section, Tables 4 and 5 represent a summary of these investigations, and we review the performance of the ML models in dealing with long-term predictions.

### 5.1. Long-Term Flood Prediction Using Single ML Methods

This section presents a comprehensive comparison on ML methods. Table 4 presents a summary of the major single ML methods used in long-term flood prediction, i.e., MLP, ANNs, SVM, and RT, followed by a comprehensive performance comparison. A revision and discussion of these methods follow, identifying the most suitable methods presented in the literature.

**Table 4.** Long-term flood prediction using single ML methods.

Modeling Technique	Reference	Flood Resource Variable	Prediction Type	Region
ANNs	[197]	Water levels	Seasonal	Sudan
ANNs	[87]	Precipitation	Monthly	Australia
BPNNs	[199]	Heavy rainfall	Seasonal	India
BPNNs vs. BFGSNN	[200]	Reservoir levels	Monthly	Turkey
BPNN vs. MLP	[201]	Discharge	Monthly	Iran
ANNs vs. HBI	[202]	Stream	Weekly	Canada
SVM vs. ANN	[203]	Streamflow	Monthly	China
RT	[204]	Floodplain forests	Annually	Australia

For seasonal flood forecasting, Elsafi [197] proposed numerous ANNs and compared the results. The water level data from different stations from 1970–1985 were selected for training, and the data from 1986–1987 were used for verification. The ANNs worked well, especially where the dataset was not complete, providing a viable choice for accurate prediction. ANNs provided the possibility of reducing the analytical costs through reducing the data analysis time that used to face in e.g., [198]. Similarly, reference [87] used ANNs to develop a prediction model for precipitation. A historical

dataset of 1900–2001 of different stations was considered and the ANN model was applied to various stations to evaluate prediction performance. The authors summarized that the ANN models offered great forecasting skills for predicting long-term evapotranspiration and precipitation. Reference [202] used an ANN model for stream assessment for long-term floods. This dataset was collected from more than 100 sites of numerous flood streams. They concluded that the ANN model, compared to Hilsenhoff's biotic index (HBI), significantly improved the prediction ability using geomorphic data. However, the ANN had generalization problems. Nevertheless, the ANN in this case proved useful to water managers.

Singh [199] used a number of BPNNs to build prediction models of heavy rains and floods. This dataset included the period of 1871–2010 on a monthly time scale. The results indicated that the BPNN models were fast and robust with simple networks, which made them great for forecasting nonlinear floods. Reference [200] aimed to better analyze nonlinear floods through modeling with BPNN and local linear regression (LLR)-based models for long-term flood forecasting. This dataset included almost two decades of rainfall, outflow, inflows, evaporation, and water level since 1988. Their evaluation concluded that LLR showed better prediction than the Broyden Fletcher Goldfarb Shanno neural network (BFGSNN) model in terms of performance and accuracy with bigger values of  $R^2$  and lower values of RMSE. However, BPNN outperformed the other methods with relatively good results. Among the ANN variations, [151] proposed a BPNN model as the most reliable ANN for long-term flood prediction. Reference [201] also compared the performances of ANNs with BPNN and MLP in the long-term prediction of flood discharge. Promising results were obtained when using MLP. However, generalization remained an issue.

Lin, Cheng, and Chau [203] applied an SVM model for estimating streamflow and reservoir inflow for a long lead time. To benchmark, they used ANNs and ARMA. The prediction models were built using monthly river flow discharges from the period of 1974–1998 for training, and 1999–2003 for testing. Through a comparison of model performance, SVM was demonstrated as a potential candidate for the prediction of long-term discharges, outshining the ANN. In a similar approach, Reference [205] proposed an SVM-based model for estimating soil moisture using remote-sensing data, and the results were compared to predictive models based on BPNN and MLR. Training was performed on the data of the period of 1998 to 2002, and testing used data from 2003 to 2005. The SVM model was shown to be more accurate and easier to build compared to BPNN and MLR. Reference [204], employed RT to model forest flood. Data from 2009–2012 at 50 sites were used for model building. The prediction of annual forest floods was reported through a combination of quantitative ground surveys, satellite imagery, hybrid machine learning tools, and future validation.

Table 5 presents a comparative analysis of single ML models for the prediction of long-term floods considering the complexity of algorithm, ease of use, running speed, accuracy, and input dataset. This table was created based on revisions that were made on articles of Table 4, as well as the accuracy analysis in Figure 9, where values of  $R^2$  and RMSE for the single ML methods were considered. The quality of the ML model prediction, in terms of speed, complexity, accuracy, and ease of use, improved continuously through the use of ensembles of ML methods, hybridization of ML methods, optimization algorithms, and/or soft computing techniques. This trend of improvement is discussed in detail in the discussion.

**Table 5.** Comparative analysis of single ML models for the prediction of long-term floods.

Modeling Technique	Complexity of Algorithm	Ease of Use	Speed	Accuracy	Input Dataset
ANN	Fairly high	Low	Fair	High	Historical
BPNN	Fairly high	Low	Fairly high	Fairly high	Historical
MLP	high	Fair	High	Fairly high	Historical
SVR	Fairly high	Low	Low	High	Historical
RT	Fair	Fair	Fair	Fairly high	Historical
SVM	Fairly high	Low	Low	High	Historical
M5 tree	Fair	Low	Fair	Fair	Historical

### 5.2. Long-Term Flood Prediction Using Hybrid ML Methods

A critical review on the long-term flood prediction using hybrid methods is presented in Table 6. Valipour, Banihabib, and Behbahani [26] used a hybrid method of autoregressive ANN integrated with sigmoid and radial activity functions. The proposed hybrid method outperformed the conventional statistical methods of ARMA and ARIMA with lower values of RMSE. They reported that ARIMA was suitable for the prediction of monthly and annual inflow, while the dynamic autoregressive ANN model with a sigmoid activity function could be used for even longer lead time. This dataset included monthly discharge from the period of 1960 to 2007.

**Table 6.** Long-term flood prediction using hybrid methods.

Modeling Technique	Reference	Flood Resource Variable	Prediction Type	Region
Autoregressive ANN vs. ARMA vs. ARIMA	[26]	River inflow	Monthly and yearly	Iran
Hybrid WNN vs. M5 model tree	[206]	Streamflow water level	Monthly	Australia
WNN vs. ANN	[207,208]	Rainfall–runoff	Monthly	Italy
WNN-BB vs. WNN vs. ANN	[50]	Streamflow	Weekly and few days	Canada
WNN vs. ANN	[25]	Urban water	Monthly	Canada
WNN vs. ANN	[209]	Peak flows	Seasonal	India
WNN vs. ANN	[210]	Rainfall	Monthly	India
WARM vs. AR	[211]	Rainfall	Yearly	Thailand
ANFIS vs. ANNs	[212]	Rainfall	Seasonal	Australia
ANFIS vs. ARMA vs. ANNs vs. SVM	[213]	Discharge	Monthly	China
ANFIS, ANNs vs. SVM vs. LLR	[214]	Streamflow	Short-term	Turkey
NLPM-ANN	[215]	Flood forecasting	Yearly	China
M-EMDSVM vs. ANN vs. SVM	[216]	Streamflow	Monthly	China
SVR-DWT-EMD	[217]	Streamflow	Monthly	China
Surrogate modeling–ML vs. ANN-Kriging model vs. ANN-PCA	[218]	Rainfall–runoff	Yearly	USA
EPS of ANNs: K-NN vs. MLP vs. MLP-PLC vs. ANNE	[219]	Streamflow	Seasonal	Canada
EEMD-ANN vs. SVM vs. ANFIS	[220]	Runoff forecast	Monthly	China
WNN vs. ANN vs. WLGP	[51]	Streamflow	Monthly	Iran

Adamowski [25] developed models based on ANN and WNN, and compared their prediction performances with statistical methods. WNN was proposed as the most accurate prediction model, as previously confirmed by Cannas et al. (2005) [207] for monthly rainfall–runoff forecasting, and also for further engineering application [208]. In a similar work, Reference [209] compared the performances of ANN and WNN for the prediction of peak flows. They also reported WNN as most reliable for simulating extreme event streams, whereby decomposition improved the results considerably. Higher levels of wavelet decomposition further improved the testing results. The statistical performance evaluation of RMSE showed considerable improvement in the testing results. Venkata Ramana [210] also combined the wavelet technique with ANN for long-term flood prediction. They considered

74 years of data for the period of 1901 to 1975. A dataset of 44 years was used for calibration, and the remainder was used for validation of the model. Their results showed a relatively lower performance for ANNs compared WNN models in modeling rainfall–runoffs. Cannas et al. [207] proposed WNN for monthly rainfall–runoff prediction, which showed significant improvement over ANNs. In a similar attempt, Kasiviswanathan, He, Sudheer, and Tay [50] used WNN and WNN–BB, which is an ensemble of WNN utilizing the block bootstrap (BB) sampling technique, to identify a robust modeling approach among ANN and WNN, by assessing accuracy and precision. This dataset included measurements from 1912 to 2013 at several flow gauge stations. The results suggested WNN–BB as a robust model for long-term streamflow prediction for longer lead times of up to one year. Tantanee et al. [211] proposed a hybrid of wavelet and autoregressive models, called WARM, which performed more effectively for long lead times. Prasad [206] proposed another similar hybrid model with the integration of WNN and iterative input selection (IIS). The hybrid model was called IIS–W-ANN, and was benchmarked with the M5 model tree. Their dataset included streamflow water level measurements from 40 years. The IIS–W-ANN hybrid model outperformed the M5 tree. This study advocated that the novel IIS–W-ANN method should be considered as an excellent flood forecasting model. Nevertheless, the model could be further optimized for better performance using optimization methods introduced in references [221–225]. In fact, such optimizers can complement IIS–W-ANN for fine-tuning the hidden-layer weights and biases for better prediction. Mekanik [212] used ANFIS to forecast seasonal rainfall. A comparison of the performance and accuracy of the ANN model and a physical model showed promising results for ANFIS. Rainfall measurements of 1900–1999 were used for training and validation, and the following decade was used for testing. The results showed that ANFIS outperformed the ANN models in all cases, comparable to Predictive Ocean Atmosphere Model for Australia (POAMA), and better than climatology. Furthermore, the study demonstrated the accuracy of ANFIS compared to global climate models. In addition, the study suggested ANFIS as an alternative tool for long-term predictions. ANFIS was reported as being easy to implement with low complexity and minimal input requirements, as well as less development time. Reference [213] compared the performances of ANFIS, ANNs, and SVM. This dataset included monthly flow data from 1953 to 2004, where the period of 2000–2004 was used for validation. ANFIS and SVM were evaluated as being better for long-term predictions. References [224,226] compared the performances of ANFIS, ANNs, and SVM for the monthly prediction of floods. The comparison results indicated that the ML models provided more accuracy than the statistical models in predicting streamflow. Furthermore, ANN and ANFIS presented more accuracy vs. SVM. However, for low-flow predictions, the SVM and ANN models outperformed ANFIS. Reference [215] proposed a modified variation of a hybrid model of NLPM–ANN to predict wetness and flood. To do so, the seasonal rainfall and wetness data of various stations were considered. The NLPM–ANN model was reported as being significantly superior to the models of previous studies. In another hybrid model, Reference [216] investigated the performance of a modified EMD–SVM (M-EMDSVM) model for long lead times, and compared its accuracy with ANN and SVM models. The M-EMDSVM model was created through modification of EMD–SVM. The evaluation results showed that the M-EMDSVM model was a better alternative to ANN, SVM, and EMD–SVM models for long lead-time streamflow prediction. The M-EMDSVM model also presented better stability, representativeness, and precision.

Zhu, Zhou, Ye, and Meng [217] contributed to the integration of ML with time-series decomposition to predict monthly streamflow through estimation and comparison of accuracy of a number of models. For that matter, they integrated SVM with discrete wavelet transform (DWT) and EMD. The hybrid models were called DWT–SVR and EMD–SVR. The results indicated that decomposition improved the accuracy of streamflow prediction, yet DWT performed even better. Further comparisons of SVR, EMD–SVR, and DWT–SVR models showed that EMD and DWT were significantly more accurate than SVR for monthly streamflow prediction.

Araghinejad [219] presented the applicability of ensembles for probabilistic flood prediction in real-life cases. He utilized the K-nearest neighbor regression for the purpose of combining individual

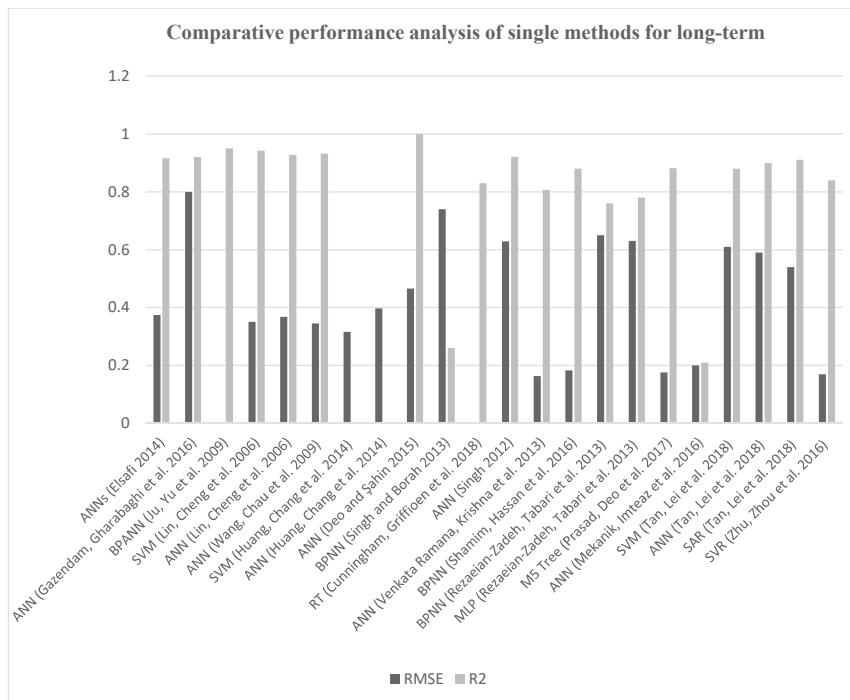
networks and improving the performance of prediction. As an EPS of ANNs, the hybrid model of K-NN was proposed to increase the generalization ability of neural networks, and was further compared with the results using MLP, MLP-PLC, and ANN. The hourly water level data of the reservoir from 132 typhoons in the period of 1971–2001 were used. The proposed EPS had a promising ability of generalization and prediction accuracy.

Bass and Bedient [218] proposed a hybrid model of surrogate-ML for long-term flood prediction suitable for TCs. The methods used included ANN integrated with principal component analysis (PCA), Kriging integrated with PC, and Kriging. The models were reported as efficient and fast to build. The results demonstrated that the methodology had an acceptable generalization ability suitable for urbanized and coastal watersheds. Reference [220] contributed to improving decomposition ensemble prediction models by developing an EEMD-ANN model for monthly prediction. The performance comparison with SVM, ANFIS, and ANNs showed a significant improvement in accuracy.

Ravansalar [51] compared the performances of the prediction models of WNN, ANN, and a novel hybrid model called wavelet linear genetic programming (WLGP) in dealing with the long-term prediction of streamflow. The results showed an accuracy of 0.87 for the WLGP model. The comparison of the performance evaluation showed that WLGP significantly increased the accuracy for the monthly approximation of peak streamflow.

## 6. Comparative Performance Analysis and Discussion

In this section, the comparative performance analysis of ML methods for long-term prediction is presented. Figure 9 represents the values of RMSE and  $R^2$  for single methods of ML, where ANNs, SVMs, and SVRs show better results. Figure 10 represents the values of RMSE and  $R^2$  for hybrid methods of ML, where decomposition and ensemble methods outperformed the more traditional methods.



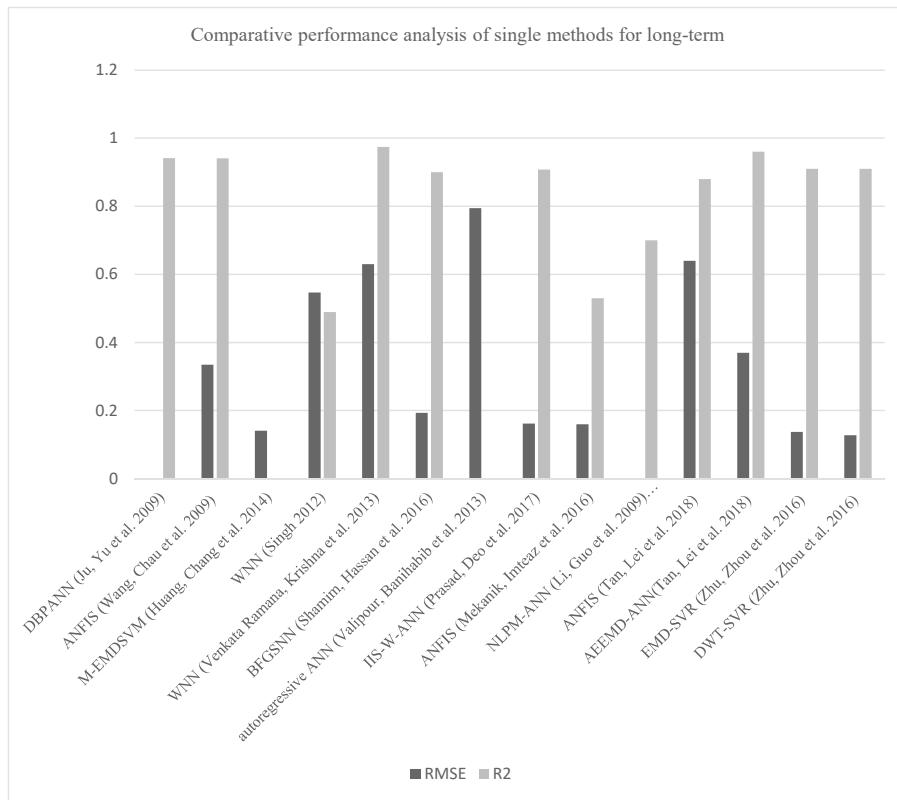
**Figure 9.** Comparative performance analysis of single methods of ML for long-term prediction.

ANNs are the most widely used ML method due to their accuracy, high fault tolerance, and powerful parallel processing in dealing with complex flood functions, especially where datasets are not complete. However, generalization remains an issue with ANN. In this context, ANFIS, MLP, and SVM performed better than ANNs. However, wavelet transforms were reported to be useful for decompositions of original time series, improving the ability of most ML methods by providing insight into datasets on various resolution levels as appropriate data pre-processing. For instance, WNNs generally produce more consistent results compared to traditional ANNs.

Either in short-term [227] or long-term rainfall-runoff modeling [50], overall, the accuracy, precision, and performance of most decomposed ML algorithms (e.g., WNN) were reported as better than those which were trained using un-decomposed time series. However, despite the achievement of WNNs, the predictions were not satisfactory for long lead times. To increase the accuracy of the longer-lead-time predictions up to one year, novel hybrids such as WARM, which is a hybrid of WNN and an autoregressive model, and wavelet multi-resolution analysis (WMRA) were proposed. In other cases, it was seen that the performance of models improved greatly through decomposition to produce cleaner inputs. For example, wavelet–neuro-fuzzy models [228] were significantly more accurate and faster than single ANFIS and ANNs. However, with an increase in the lead time, the uncertainty in prediction increased. Thus, the evaluation of model precision should come into consideration in future studies.

Data decomposition methods, e.g., autoregressive, wavelet transforms, wavelet–autoregressive, DWT, IIS, and EMD, contributed highly to developing hybrid methods for longer prediction lead time, good stability, great representativeness, and higher accuracy. These data decomposition methods were integrated with ANNs, SVM, WNN, and FR, and they are expected to gain more popularity among researchers. The other trend in improvement of prediction accuracy and generalization capability involves EPS. In fact, recent ensemble methods contributed to good improvements in speed, accuracy, and generalization. The EPS of ANNs and WNNs, using BB sampling, genetic programming, simple average, stop training, Bayesian, data fusion, regression, and other soft computing techniques, showed promising results and better performances than traditional ML methods. In ensembles, however, it is noted that human decision as the input variable provided superior performance than models without this important input. However, the most significant hybrid models were novel decomposition–ensemble prediction models suitable for monthly prediction. Their performance comparisons with SVM, ANFIS, and ANNs showed significant improvements in accuracy and generalization. Figure 10 represents the comparative performance analysis of hybrid methods of ML for short-term prediction. Here, it is also worth mentioning the importance of further signal processing techniques (e.g., Reference [228]) for both long-term and short-term floods.

This paper suggests that the drawbacks to major ML methods in terms of accuracy, uncertainty, performance, and robustness were improved through the hybridization of ML methods, as well as using an ensemble variation of the ML method. It is expected that this trend represents the future horizon of flood prediction.



**Figure 10.** Comparative performance analysis of hybrid methods of ML for short-term prediction.

## 7. Conclusions

The current state of ML modeling for flood prediction is quite young and in the early stage of advancement. This paper presents an overview of machine learning models used in flood prediction, and develops a classification scheme to analyze the existing literature. The survey represents the performance analysis and investigation of more than 6000 articles. Among them, we identified 180 original and influential articles where the performance and accuracy of at least two machine learning models were compared. To do so, the prediction models were classified into two categories according to lead time, and further divided into categories of hybrid and single methods. The state of the art of these classes was discussed and analyzed in detail, considering the performance comparison of the methods available in the literature. The performance of the methods was evaluated in terms of  $R^2$  and RMSE, in addition to the generalization ability, robustness, computation cost, and speed. Despite the promising results already reported in implementing the most popular machine learning methods, e.g., ANNs, SVM, SVR, ANFIS, WNN, and DTs, there was significant research and experimentation for further improvement and advancement. In this context, there were four major trends reported in the literature for improving the quality of prediction. The first was novel hybridization, either through the integration of two or more machine learning methods or the integration of a machine learning method(s) with more conventional means, and/or soft computing. The second was the use of data decomposition techniques for the purpose of improving the quality of the dataset, which highly contributed in improving the accuracy of prediction. The third was the use of an ensemble of methods, which dramatically increased the generalization ability of the models and decreased the uncertainty

of prediction. The fourth was the use of add-on optimizer algorithms to improve the quality of machine learning algorithms, e.g., for better tuning the ANNs to reach optimal neuronal architectures. It is expected that, through these four key technologies, flood prediction will witness significant improvements for both short-term and long-term predictions. Surely, the advancement of these novel ML methods depends highly on the proper usage of soft computing techniques in designing novel learning algorithms. This fact was discussed in the paper, and the soft computing techniques were introduced as the main contributors in developing hybrid ML methods of the future.

Here, it is also worth mentioning that the multidisciplinary nature of this work was the most challenging difficulty to overcome in this paper. Having contributions from the coauthors of both realms of ML and hydrology was the key to success. Furthermore, the novel search methodology and the creative taxonomy and classification of the ML methods led to the original achievement of the paper.

For future work, conducting a survey on spatial flood prediction using machine learning models is highly encouraged. This important aspect of flood prediction was excluded from our paper due to the nature of modeling methodologies and the datasets used in predicting the location of floods. Nevertheless, the recent advancements in machine learning models for spatial flood analysis revolutionized this particular realm of flood forecasting, which requires separate investigation.

**Author Contributions:** P.O., the machine learning expert, contributed to investigation, methodology, supervision, communication, resources, data curation, and project management. A.M. contributed to the original draft preparation, data collection, formal data analysis, investigation, critical review, and final revisions, K.-w.C., the hydrology expert, contributed to supervision, validation, revision, discussion, resources, improvement, and advice.

**Funding:** This research was funded by the Norwegian University of Science and Technology AI Lab and the European Research Consortium for Informatics and Mathematics (ERCIM).

**Acknowledgments:** Amir Mosavi contributed to this research during his tenure through an ERCIM Alain Bensoussan Fellowship Program under supervision of Pinar Ozturk.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Nomenclatures

WMO	World meteorological organization
GCM	Global circulation models
SPOTA	Seasonal Pacific Ocean temperature analysis
ANN	Artificial neural networks
POTA	Pacific Ocean temperature analysis
QPE	Quantitative precipitation estimation
CLIM	Climatology average method
EOF	Empirical orthogonal function
MLR	Multiple linear regressions
QPF	Quantitative precipitation forecasting
MNLR	Multiple nonlinear regressions
ML	Machine learning
MLR	Multiple linear regression
ANN	Neural networks
WNN	Wavelet-based neural network
ARIMA	Auto regressive integrated moving average
USGS	United States Geological Survey
FFA	Flood frequency analyses
QRT	Quantile regression techniques
SPOTA	Seasonal Pacific Ocean temperature analysis
SVM	Support vector machines
LS-SVM	Least-square support vector machines
AI	Artificial intelligence

VRM	Vector Regression Machine
FFNN	Feed-forward neural network
FBNN	Feed-backward networks
MLP	Multilayer perceptron
ANFIS	Adaptive neuro-fuzzy inference system
BPNN	Backpropagation neural network
SVR	Support vector regression
LR	Linear regression
FIS	Fuzzy inference system
CART	Classification and regression tree
LMT	Logistic model trees
NWP	Numerical weather prediction
NBT	Naïve Bayes trees
ARMA	Autoregressive moving averaging
REPT	Reduced-error pruning trees
DT	Decision tree
ELM	Extreme learning machine
EPS	Ensemble prediction systems
SNIP	Source normalized impact per paper
SRM	Structural risk minimization
AR	Autoregressive
SJR	SCImago journal rank
ARMAX	Linear autoregressive moving average with exogenous inputs
LMT	Logistic model trees
ARMA	Autoregressive moving averaging
ADT	Alternating decision trees
NARX network	Nonlinear autoregressive network with exogenous inputs
RMSE	Root-mean-square error
RFFA	Regional flood frequency analysis
NLR	Nonlinear regression
AR	Autoregressive
WARM	Wavelet autoregressive model
NLR-R	Nonlinear regression with regionalization approach
E	Nash Sutcliffe index
FR	Frequency ratio
SOM	Self-organizing map
CHIM	Cluster-based hybrid inundation model
FFRM	Flash flood routing model
KGE	Kling-Gupta efficiency
AME	ANN-based monsoon rainfall enhancement
SSNN	State-space neural network
SSL	Suspended sediment load
NSE	Nash–Sutcliffe efficiency
E-CHAID	Exhaustive CHAID
CHAID	Chi-squared automatic interaction detector
CLIM	Climatology average model
HEC-HMS	Hydrologic engineering left–hydrologic modeling system
SOM	Self-organizing map
PBIAS	Percent bias
NLPM	Nonlinear perturbation model
RF	Rotation forest
KSOFM-NNM	Kohonen self-organizing feature maps neural networks model
DBP	Division-based backpropagation
DBPANN	DBP neural network

NLPM-ANN	Nonlinear perturbation model based on neural network
GRNNM	Generalized regression neural networks model
IIS	Iterative input selection
EEMD	Ensemble empirical mode decomposition
ANNE	Artificial neural network ensembles
DWT	Discrete wavelet transform
SFF	Seasonal flood forecasting
MP	Water monitoring points
WBANN	Wavelet-bootstrap-ANN
HBI	Hilsenhoff's biotic index
RT	Regression trees
EMD	Empirical mode decomposition
LLR	Local linear regression
BFGS	Broyden Fletcher Goldfarb Shanno
M-EMD	Modified empirical mode decomposition
IIS	Iterative input selection
SAR	Seasonal first-order autoregressive
BFGSNN	Broyden Fletcher Goldfarb Shanno neural network
GRNN	Artificial neural networks including generalized regression network
T-S	Takagi–Sugeno
WLGP	Wavelet linear genetic programming
E	Nash coefficients
TSC-T-S	Clustering based Takagi–Sugeno
TCs	Tropical cyclones
PCA	Principal component analysis

## References

1. Danso-Amoako, E.; Scholz, M.; Kalimeris, N.; Yang, Q.; Shao, J. Predicting dam failure risk for sustainable flood retention basins: A generic case study for the wider greater manchester area. *Comput. Environ. Urban Syst.* **2012**, *36*, 423–433. [[CrossRef](#)]
2. Xie, K.; Ozbay, K.; Zhu, Y.; Yang, H. Evacuation zone modeling under climate change: A data-driven method. *J. Infrastruct. Syst.* **2017**, *23*, 04017013. [[CrossRef](#)]
3. Pitt, M. *Learning Lessons from the 2007 Floods*; Cabinet Office: London, UK, 2008.
4. Lohani, A.K.; Goel, N.; Bhatia, K. Improving real time flood forecasting using fuzzy inference system. *J. Hydrol.* **2014**, *509*, 25–41. [[CrossRef](#)]
5. Mosavi, A.; Bathla, Y.; Varkonyi-Koczy, A. Predicting the Future Using Web Knowledge: State of the Art Survey. In *Recent Advances in Technology Research and Education*; Springer: Cham, Switzerland, 2017; pp. 341–349.
6. Zhao, M.; Hendon, H.H. Representation and prediction of the indian ocean dipole in the poama seasonal forecast model. *Q. J. R. Meteorol. Soc.* **2009**, *135*, 337–352. [[CrossRef](#)]
7. Borah, D.K. Hydrologic procedures of storm event watershed models: A comprehensive review and comparison. *Hydrol. Process.* **2011**, *25*, 3472–3489. [[CrossRef](#)]
8. Costabile, P.; Costanzo, C.; Macchione, F. A storm event watershed model for surface runoff based on 2D fully dynamic wave equations. *Hydrol. Process.* **2013**, *27*, 554–569. [[CrossRef](#)]
9. Cea, L.; Garrido, M.; Puertas, J. Experimental validation of two-dimensional depth-averaged models for forecasting rainfall-runoff from precipitation data in urban areas. *J. Hydrol.* **2010**, *382*, 88–102. [[CrossRef](#)]
10. Fernández-Pato, J.; Caviedes-Voullième, D.; García-Navarro, P. Rainfall/runoff simulation with 2D full shallow water equations: Sensitivity analysis and calibration of infiltration parameters. *J. Hydrol.* **2016**, *536*, 496–513. [[CrossRef](#)]
11. Caviedes-Voullième, D.; García-Navarro, P.; Murillo, J. Influence of mesh structure on 2D full shallow water equations and SCS curve number simulation of rainfall/runoff events. *J. Hydrol.* **2012**, *448*, 39–59. [[CrossRef](#)]
12. Costabile, P.; Costanzo, C.; Macchione, F. Comparative analysis of overland flow models using finite volume schemes. *J. Hydroinform.* **2012**, *14*, 122. [[CrossRef](#)]

13. Xia, X.; Liang, Q.; Ming, X.; Hou, J. An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations. *Water Resour. Res.* **2017**, *53*, 3730–3759. [[CrossRef](#)]
14. Liang, X.; Lettenmaier, D.P.; Wood, E.F.; Burges, S.J. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *J. Geophys. Res. Atmos.* **1994**, *99*, 14415–14428. [[CrossRef](#)]
15. Costabile, P.; Macchione, F. Enhancing river model set-up for 2-D dynamic flood modelling. *Environ. Model. Softw.* **2015**, *67*, 89–107. [[CrossRef](#)]
16. Nayak, P.; Sudheer, K.; Rangan, D.; Ramasastri, K. Short-term flood forecasting with a neurofuzzy model. *Water Resour. Res.* **2005**, *41*. [[CrossRef](#)]
17. Kim, B.; Sanders, B.F.; Famiglietti, J.S.; Guinot, V. Urban flood modeling with porous shallow-water equations: A case study of model errors in the presence of anisotropic porosity. *J. Hydrol.* **2015**, *523*, 680–692. [[CrossRef](#)]
18. Van den Honert, R.C.; McAneney, J. The 2011 brisbane floods: Causes, impacts and implications. *Water* **2011**, *3*, 1149–1173. [[CrossRef](#)]
19. Lee, T.H.; Georgakakos, K.P. Operational rainfall prediction on meso- $\gamma$  scales for hydrologic applications. *Water Resour. Res.* **1996**, *32*, 987–1003. [[CrossRef](#)]
20. Shrestha, D.; Robertson, D.; Wang, Q.; Pagano, T.; Hapuarachchi, H. Evaluation of numerical weather prediction model precipitation forecasts for short-term streamflow forecasting purpose. *Hydrol. Earth Syst. Sci.* **2013**, *17*, 1913–1931. [[CrossRef](#)]
21. Bellos, V.; Tsakiris, G. A hybrid method for flood simulation in small catchments combining hydrodynamic and hydrological techniques. *J. Hydrol.* **2016**, *540*, 331–339. [[CrossRef](#)]
22. Bout, B.; Jetten, V. The validity of flow approximations when simulating catchment-integrated flash floods. *J. Hydrol.* **2018**, *556*, 674–688. [[CrossRef](#)]
23. Costabile, P.; Macchione, F.; Natale, L.; Petaccia, G. Flood mapping using lidar dem. Limitations of the 1-D modeling highlighted by the 2-D approach. *Nat. Hazards* **2015**, *77*, 181–204. [[CrossRef](#)]
24. Valipour, M.; Banihabib, M.E.; Behbahani, S.M.R. Parameters estimate of autoregressive moving average and autoregressive integrated moving average models and compare their ability for inflow forecasting. *J. Math. Stat.* **2012**, *8*, 330–338.
25. Adamowski, J.; Fung Chan, H.; Prasher, S.O.; Ozga-Zielinski, B.; Sliusarieva, A. Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in montreal, Canada. *Water Resour. Res.* **2012**, *48*. [[CrossRef](#)]
26. Valipour, M.; Banihabib, M.E.; Behbahani, S.M.R. Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *J. Hydrol.* **2013**, *476*, 433–441. [[CrossRef](#)]
27. Chow, V.T.; Maidment, D.R.; Larry, W. *Mays. Applied hydrology*; International Edition; MacGraw-Hill, Inc.: New York, NY, USA, 1988; p. 149.
28. Aziz, K.; Rahman, A.; Fang, G.; Shrestha, S. Application of artificial neural networks in regional flood frequency analysis: A case study for australia. *Stoch. Environ. Res. Risk Assess.* **2014**, *28*, 541–554. [[CrossRef](#)]
29. Kroll, C.N.; Vogel, R.M. Probability distribution of low streamflow series in the united states. *J. Hydrol. Eng.* **2002**, *7*, 137–146. [[CrossRef](#)]
30. Mackey, B.P.; Krishnamurti, T. Ensemble forecast of a typhoon flood event. *Weather Forecast.* **2001**, *16*, 399–415. [[CrossRef](#)]
31. Haddad, K.; Rahman, A. Regional flood frequency analysis in eastern australia: Bayesian GLS regression-based methods within fixed region and ROI framework–quantile regression vs. Parameter regression technique. *J. Hydrol.* **2012**, *430*, 142–161. [[CrossRef](#)]
32. Thompson, S.A. *Hydrology for Water Management*; CRC Press: Boca Raton, FL, USA, 2017.
33. Kerkhoven, E.; Gan, T.Y. A modified ISBA surface scheme for modeling the hydrology of Athabasca river basin with GCM-scale data. *Adv. Water Resour.* **2006**, *29*, 808–826. [[CrossRef](#)]
34. Burnash, R.J.; Ferral, R.L.; McGuire, R.A. *A Generalized Streamflow Simulation System, Conceptual Modeling for Digital Computers*; Stanford University: Stanford, CA, USA, 1973.
35. Yamazaki, D.; Kanae, S.; Kim, H.; Oki, T. A physically based description of floodplain inundation dynamics in a global river routing model. *Water Resour. Res.* **2011**, *47*. [[CrossRef](#)]

36. Fawcett, R.; Stone, R. A comparison of two seasonal rainfall forecasting systems for Australia. *Aust. Meteorol. Oceanogr. J.* **2010**, *60*, 15–24. [[CrossRef](#)]
37. Mekanik, F.; Imteaz, M.; Gato-Trinidad, S.; Elmahdi, A. Multiple regression and artificial neural network for long-term rainfall forecasting using large scale climate modes. *J. Hydrol.* **2013**, *503*, 11–21. [[CrossRef](#)]
38. Mosavi, A.; Rabczuk, T.; Varkonyi-Koczy, A.R. Reviewing the novel machine learning tools for materials design. In *Recent Advances in Technology Research and Education*; Springer: Cham, Switzerland, 2017; pp. 50–58.
39. Abbot, J.; Marohasy, J. Input selection and optimisation for monthly rainfall forecasting in Queensland, Australia, using artificial neural networks. *Atmos. Res.* **2014**, *138*, 166–178. [[CrossRef](#)]
40. Fox, N.I.; Wikle, C.K. A bayesian quantitative precipitation nowcast scheme. *Weather Forecast.* **2005**, *20*, 264–275. [[CrossRef](#)]
41. Merz, B.; Hall, J.; Disse, M.; Schumann, A. Fluvial flood risk management in a changing world. *Nat. Hazards Earth Syst. Sci.* **2010**, *10*, 509–527. [[CrossRef](#)]
42. Xu, Z.; Li, J. Short-term inflow forecasting using an artificial neural network model. *Hydrol. Process.* **2002**, *16*, 2423–2439. [[CrossRef](#)]
43. Ortiz-García, E.; Salcedo-Sanz, S.; Casanova-Mateo, C. Accurate precipitation prediction with support vector classifiers: A study including novel predictive variables and observational data. *Atmos. Res.* **2014**, *139*, 128–136. [[CrossRef](#)]
44. Kim, S.; Matsumi, Y.; Pan, S.; Mase, H. A real-time forecast model using artificial neural network for after-runner storm surges on the Tottori Coast, Japan. *Ocean Eng.* **2016**, *122*, 44–53. [[CrossRef](#)]
45. Mosavi, A.; Edalatifar, M. A. Edalatifar, M. A Hybrid Neuro-Fuzzy Algorithm for Prediction of Reference Evapotranspiration. In *Recent Advances in Technology Research and Education*; Springer: Cham, Switzerland, 2018; pp. 235–243.
46. Dineva, A.; Várkonyi-Kóczy, A.R.; Tar, J.K. Fuzzy expert system for automatic wavelet shrinkage procedure selection for noise suppression. In Proceedings of the 2014 IEEE 18th International Conference on Intelligent Engineering Systems (INES), Tihany, Hungary, 3–5 July 2014; pp. 163–168.
47. Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [[CrossRef](#)]
48. Gizaw, M.S.; Gan, T.Y. Regional flood frequency analysis using support vector regression under historical and future climate. *J. Hydrol.* **2016**, *538*, 387–398. [[CrossRef](#)]
49. Taherei Ghazvini, P.; Hassanpour Darvishi, H.; Mosavi, A.; Yusof, K.B.W.; Alizamir, M.; Shamshirband, S.; Chau, K.W. Sugarcane growth prediction based on meteorological parameters using extreme learning machine and artificial neural network. *Eng. Appl. Comput. Fluid Mech.* **2018**, *12*, 738–749. [[CrossRef](#)]
50. Kasiviswanathan, K.; He, J.; Sudheer, K.; Tay, J.-H. Potential application of wavelet neural network ensemble to forecast streamflow for flood management. *J. Hydrol.* **2016**, *536*, 161–173. [[CrossRef](#)]
51. Ravansalar, M.; Rajaei, T.; Kisi, O. Wavelet-linear genetic programming: A new approach for modeling monthly streamflow. *J. Hydrol.* **2017**, *549*, 461–475. [[CrossRef](#)]
52. Mosavi, A.; Rabczuk, T. Learning and intelligent optimization for material design innovation. In *Learning and Intelligent Optimization*; Springer: Cham, Switzerland, 2017; pp. 358–363.
53. Dandagala, S.; Reddy, M.S.; Murthy, D.S.; Nagaraj, G. Artificial neural networks applications in groundwater hydrology—A review. *Artif. Intell. Syst. Mach. Learn.* **2017**, *9*, 182–187.
54. Deka, P.C. Support vector machine applications in the field of hydrology: A review. *Appl. Soft Comput.* **2014**, *19*, 372–386.
55. Fotovatikhah, F.; Herrera, M.; Shamshirband, S.; Chau, K.-W.; Faizollahzadeh Ardabili, S.; Piran, M.J. Survey of computational intelligence as basis to big flood management: Challenges, research directions and future work. *Eng. Appl. Comput. Fluid Mech.* **2018**, *12*, 411–437. [[CrossRef](#)]
56. Faizollahzadeh Ardabili, S.; Najafi, B.; Alizamir, M.; Mosavi, A.; Shamshirband, S.; Rabczuk, T. Using SVM-RSM and ELM-RSM Approaches for Optimizing the Production Process of Methyl and Ethyl Esters. *Energies* **2018**, *11*, 2889. [[CrossRef](#)]
57. Tsai, L.T.; Yang, C.-C. Improving measurement invariance assessments in survey research with missing data by novel artificial neural networks. *Expert Syst. Appl.* **2012**, *39*, 10456–10464. [[CrossRef](#)]
58. Sivapalan, M.; Blöschl, G.; Merz, R.; Gutknecht, D. Linking flood frequency to long-term water balance: Incorporating effects of seasonality. *Water Resour. Res.* **2005**, *41*. [[CrossRef](#)]

59. Maier, H.R.; Jain, A.; Dandy, G.C.; Sudheer, K.P. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environ. Model. Softw.* **2010**, *25*, 891–909. [[CrossRef](#)]
60. Lafdani, E.K.; Nia, A.M.; Pahlavanravi, A.; Ahmadi, A.; Jajarmizadeh, M. Research article daily rainfall-runoff prediction and simulation using ANN, ANFIS and conceptual hydrological MIKE11/NAM models. *Int. J. Eng. Technol.* **2013**, *1*, 32–50.
61. Collier, C. Flash flood forecasting: What are the limits of predictability? *Q. J. R. Meteorol. Soc.* **2007**, *133*, 3–23. [[CrossRef](#)]
62. Seo, D.-J.; Breidenbach, J. Real-time correction of spatially nonuniform bias in radar rainfall data using rain gauge measurements. *J. Hydrometeorol.* **2002**, *3*, 93–111. [[CrossRef](#)]
63. Grecu, M.; Krajewski, W. A large-sample investigation of statistical procedures for radar-based short-term quantitative precipitation forecasting. *J. Hydrol.* **2000**, *239*, 69–84. [[CrossRef](#)]
64. Maddox, R.A.; Zhang, J.; Gourley, J.J.; Howard, K.W. Weather radar coverage over the contiguous united states. *Weather Forecast.* **2002**, *17*, 927–934. [[CrossRef](#)]
65. Campolo, M.; Andreussi, P.; Soldati, A. River flood forecasting with a neural network model. *Water Resour. Res.* **1999**, *35*, 1191–1197. [[CrossRef](#)]
66. Prakash, O.; Sudheer, K.; Srinivasan, K. Improved higher lead time river flow forecasts using sequential neural network with error updating. *J. Hydrol. Hydromech.* **2014**, *62*, 60–74. [[CrossRef](#)]
67. Shu, C.; Ouarda, T. Regional flood frequency analysis at ungauged sites using the adaptive neuro-fuzzy inference system. *J. Hydrol.* **2008**, *349*, 31–43. [[CrossRef](#)]
68. Ashrafi, M.; Chua, L.H.C.; Quek, C.; Qin, X. A fully-online neuro-fuzzy model for flow forecasting in basins with limited data. *J. Hydrol.* **2017**, *545*, 424–435. [[CrossRef](#)]
69. Yu, P.-S.; Yang, T.-C.; Chen, S.-Y.; Kuo, C.-M.; Tseng, H.-W. Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting. *J. Hydrol.* **2017**, *552*, 92–104. [[CrossRef](#)]
70. Nourani, V.; Baghanam, A.H.; Adamowski, J.; Kisi, O. Applications of hybrid wavelet–artificial intelligence models in hydrology: A review. *J. Hydrol.* **2014**, *514*, 358–377. [[CrossRef](#)]
71. Zadeh, M.R.; Amin, S.; Khalili, D.; Singh, V.P. Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions. *Water Resour. Manag.* **2010**, *24*, 2673–2688. [[CrossRef](#)]
72. Li, L.; Xu, H.; Chen, X.; Simonovic, S. Streamflow forecast and reservoir operation performance assessment under climate change. *Water Resour. Manag.* **2010**, *24*, 83. [[CrossRef](#)]
73. Wu, C.; Chau, K.-W. Data-driven models for monthly streamflow time series prediction. *Eng. Appl. Artif. Intell.* **2010**, *23*, 1350–1367. [[CrossRef](#)]
74. Sulaiman, J.; Wahab, S.H. Heavy rainfall forecasting model using artificial neural network for flood prone area. In *It Convergence and Security 2017*; Springer: Singapore, 2018; pp. 68–76.
75. Kar, A.K.; Lohani, A.K.; Goel, N.K.; Roy, G.P. Development of flood forecasting system using statistical and ANN techniques in the downstream catchment of mahanadi basin, india. *J. Water Resour. Prot.* **2010**, *2*, 880.
76. Jain, A.; Prasad Indurthy, S. Closure to “comparative analysis of event-based rainfall-runoff modeling techniques—Deterministic, statistical, and artificial neural networks” by ASHU JAIN and SKV prasad indurthy. *J. Hydrol. Eng.* **2004**, *9*, 551–553. [[CrossRef](#)]
77. Lohani, A.; Kumar, R.; Singh, R. Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. *J. Hydrol.* **2012**, *442*, 23–35. [[CrossRef](#)]
78. Tanty, R.; Desmukh, T.S. Application of artificial neural network in hydrology—A review. *Int. J. Eng. Technol. Res.* **2015**, *4*, 184–188.
79. Kişi, O. Streamflow forecasting using different artificial neural network algorithms. *J. Hydrol. Eng.* **2007**, *12*, 532–539. [[CrossRef](#)]
80. Shamseldin, A.Y. Artificial neural network model for river flow forecasting in a developing country. *J. Hydroinform.* **2010**, *12*, 22–35. [[CrossRef](#)]
81. Badrzadeh, H.; Sarukkalige, R.; Jayawardena, A. Impact of multi-resolution analysis of artificial intelligence models inputs on multi-step ahead river flow forecasting. *J. Hydrol.* **2013**, *507*, 75–85. [[CrossRef](#)]
82. Smith, J.; Eli, R.N. Neural-network models of rainfall-runoff process. *J. Water Resour. Plan. Manag.* **1995**, *121*, 499–508. [[CrossRef](#)]
83. Taormina, R.; Chau, K.-W.; Sethi, R. Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice Lagoon. *Eng. Appl. Artif. Intell.* **2012**, *25*, 1670–1676. [[CrossRef](#)]

84. Thirumalaiah, K.; Deo, M. River stage forecasting using artificial neural networks. *J. Hydrol. Eng.* **1998**, *3*, 26–32. [[CrossRef](#)]
85. Panagoulia, D. Artificial neural networks and high and low flows in various climate regimes. *Hydrol. Sci. J.* **2006**, *51*, 563–587. [[CrossRef](#)]
86. Panagoulia, D.; Tsekouras, G.; Kousouris, G. A multi-stage methodology for selecting input variables in annual forecasting of river flows. *Glob. Nest J.* **2017**, *19*, 49–57.
87. Deo, R.C.; Şahin, M. Application of the artificial neural network model for prediction of monthly standardized precipitation and evapotranspiration index using hydrometeorological parameters and climate indices in Eastern Australia. *Atmos. Res.* **2015**, *161*, 65–81. [[CrossRef](#)]
88. Coulibaly, P.; Dibike, Y.B.; Anctil, F. Downscaling precipitation and temperature with temporal neural networks. *J. Hydrometeorol.* **2005**, *6*, 483–496. [[CrossRef](#)]
89. Schoof, J.T.; Pryor, S. Downscaling temperature and precipitation: A comparison of regression-based methods and artificial neural networks. *Int. J. Climatol.* **2001**, *21*, 773–790. [[CrossRef](#)]
90. Hassan, Z.; Shamsudin, S.; Harun, S.; Malek, M.A.; Hamidon, N. Suitability of ANN applied as a hydrological model coupled with statistical downscaling model: A case study in the northern area of peninsular Malaysia. *Environ. Earth Sci.* **2015**, *74*, 463–477. [[CrossRef](#)]
91. Zhang, J.-S.; Xiao, X.-C. Predicting chaotic time series using recurrent neural network. *Chin. Phys. Lett.* **2000**, *17*, 88. [[CrossRef](#)]
92. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
93. Lima, A.R.; Cannon, A.J.; Hsieh, W.W. Forecasting daily streamflow using online sequential extreme learning machines. *J. Hydrol.* **2016**, *537*, 431–443. [[CrossRef](#)]
94. Yaseen, Z.M.; Jaafar, O.; Deo, R.C.; Kisi, O.; Adamowski, J.; Quilty, J.; El-Shafie, A. Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq. *J. Hydrol.* **2016**, *542*, 603–614. [[CrossRef](#)]
95. Sahoo, G.; Ray, C. Flow forecasting for a Hawaii stream using rating curves and neural networks. *J. Hydrol.* **2006**, *317*, 63–80. [[CrossRef](#)]
96. Kim, S.; Singh, V.P. Flood forecasting using neural computing techniques and conceptual class segregation. *JAWRA J. Am. Water Resour. Assoc.* **2013**, *49*, 1421–1435. [[CrossRef](#)]
97. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533. [[CrossRef](#)]
98. Riad, S.; Mania, J.; Bouchaou, L.; Najjar, Y. Rainfall-runoff model using an artificial neural network approach. *Math. Comput. Model.* **2004**, *40*, 839–846. [[CrossRef](#)]
99. Senthil Kumar, A.; Sudheer, K.; Jain, S.; Agarwal, P. Rainfall-runoff modelling using artificial neural networks: Comparison of network types. *Hydrol. Process. Int. J.* **2005**, *19*, 1277–1291. [[CrossRef](#)]
100. Zadeh, L.A. Soft computing and fuzzy logic. In *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi a Zadeh*; World Scientific: Singapore, 1996; pp. 796–804.
101. Choubin, B.; Khalighi-Sigaroodi, S.; Malekian, A.; Ahmad, S.; Attarod, P. Drought forecasting in a semi-arid watershed using climate signals: A neuro-fuzzy modeling approach. *J. Mt. Sci.* **2014**, *11*, 1593–1605. [[CrossRef](#)]
102. Choubin, B.; Khalighi-Sigaroodi, S.; Malekian, A.; Kişi, Ö. Multiple linear regression, multi-layer perceptron network and adaptive neuro-fuzzy inference system for forecasting precipitation based on large-scale climate signals. *Hydrol. Sci. J.* **2016**, *61*, 1001–1009. [[CrossRef](#)]
103. Bogardi, I.; Duckstein, L. The fuzzy logic paradigm of risk analysis. In *Risk-Based Decisionmaking in Water Resources X*; American Society of Civil Engineers: Reston, VA, USA, 2003; pp. 12–22.
104. See, L.; Openshaw, S. A hybrid multi-model approach to river level forecasting. *Hydrol. Sci. J.* **2000**, *45*, 523–536. [[CrossRef](#)]
105. Tiwari, M.K.; Chatterjee, C. Development of an accurate and reliable hourly flood forecasting model using wavelet-bootstrap-ANN (WBANN) hybrid approach. *J. Hydrol.* **2010**, *394*, 458–470. [[CrossRef](#)]
106. Guimarães Santos, C.A.; da Silva, G.B.L. Daily streamflow forecasting using a wavelet transform and artificial neural network hybrid models. *Hydrol. Sci. J.* **2014**, *59*, 312–324. [[CrossRef](#)]

107. Supratid, S.; Aribarg, T.; Supharatid, S. An integration of stationary wavelet transform and nonlinear autoregressive neural network with exogenous input for baseline and future forecasting of reservoir inflow. *Water Resour. Manag.* **2017**, *31*, 4023–4043. [[CrossRef](#)]
108. Shoaib, M.; Shamseldin, A.Y.; Melville, B.W. Comparative study of different wavelet based neural network models for rainfall-runoff modeling. *J. Hydrol.* **2014**, *515*, 47–58. [[CrossRef](#)]
109. Dubossarsky, E.; Friedman, J.H.; Ormerod, J.T.; Wand, M.P. Wavelet-based gradient boosting. *Stat. Comput.* **2016**, *26*, 93–105. [[CrossRef](#)]
110. Partal, T. Wavelet regression and wavelet neural network models for forecasting monthly streamflow. *J. Water Clim. Chang.* **2017**, *8*, 48–61. [[CrossRef](#)]
111. Shafaei, M.; Kisi, O. Predicting river daily flow using wavelet-artificial neural networks based on regression analyses in comparison with artificial neural networks and support vector machine models. *Neural Comput. Appl.* **2017**, *28*, 15–28. [[CrossRef](#)]
112. Kumar, S.; Tiwari, M.K.; Chatterjee, C.; Mishra, A. Reservoir inflow forecasting using ensemble models based on neural networks, wavelet analysis and bootstrap method. *Water Resour. Manag.* **2015**, *29*, 4863–4883. [[CrossRef](#)]
113. Seo, Y.; Kim, S.; Kisi, O.; Singh, V.P. Daily water level forecasting using wavelet decomposition and artificial intelligence techniques. *J. Hydrol.* **2015**, *520*, 224–243. [[CrossRef](#)]
114. Sudhishri, S.; Kumar, A.; Singh, J.K. Comparative evaluation of neural network and regression based models to simulate runoff and sediment yield in an outer Himalayan watershed. *J. Agric. Sci. Technol.* **2016**, *18*, 681–694.
115. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [[CrossRef](#)]
116. Vapnik, V.; Mukherjee, S. Support vector method for multivariate density estimation. *Adv. Neural Inf. Process. Syst.* **2000**, *4*, 659–665.
117. Li, S.; Ma, K.; Jin, Z.; Zhu, Y. A new flood forecasting model based on SVM and boosting learning algorithms. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 1343–1348.
118. Dehghani, M.; Saghafian, B.; Nasiri Saleh, F.; Farokhnia, A.; Noori, R. Uncertainty analysis of streamflow drought forecast using artificial neural networks and Monte-Carlo simulation. *Int. J. Climatol.* **2014**, *34*, 1169–1180. [[CrossRef](#)]
119. Dibike, Y.B.; Velickov, S.; Solomatine, D.; Abbott, M.B. Model induction with support vector machines: Introduction and applications. *J. Comput. Civ. Eng.* **2001**, *15*, 208–216. [[CrossRef](#)]
120. Nayak, M.A.; Ghosh, S. Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier. *Theor. Appl. Climatol.* **2013**, *114*, 583–603. [[CrossRef](#)]
121. Granata, F.; Gargano, R.; de Marinis, G. Support vector regression for rainfall-runoff modeling in urban drainage: A comparison with the EPA's storm water management model. *Water* **2016**, *8*, 69. [[CrossRef](#)]
122. Gong, Y.; Zhang, Y.; Lan, S.; Wang, H. A comparative study of artificial neural networks, support vector machines and adaptive neuro fuzzy inference system for forecasting groundwater levels near lake okeechobee, Florida. *Water Resour. Manag.* **2016**, *30*, 375–391. [[CrossRef](#)]
123. Jajarmizadeh, M.; Lafdani, E.K.; Harun, S.; Ahmadi, A. Application of SVM and swat models for monthly streamflow prediction, a case study in South of Iran. *KSCE J. Civ. Eng.* **2015**, *19*, 345–357. [[CrossRef](#)]
124. Bao, Y.; Xiong, T.; Hu, Z. Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing* **2014**, *129*, 482–493. [[CrossRef](#)]
125. Bray, M.; Han, D. Identification of support vector machines for runoff modelling. *J. Hydroinform.* **2004**, *6*, 265–280. [[CrossRef](#)]
126. Tehrany, M.S.; Pradhan, B.; Mansor, S.; Ahmad, N. Flood susceptibility assessment using GIS-based support vector machine model with different kernel types. *Catena* **2015**, *125*, 91–101. [[CrossRef](#)]
127. Kisi, O.; Parmar, K.S. Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution. *J. Hydrol.* **2016**, *534*, 104–112. [[CrossRef](#)]
128. Liong, S.Y.; Sivapragasam, C. Flood stage forecasting with support vector machines. *JAWRA J. Am. Water Resour. Assoc.* **2002**, *38*, 173–186. [[CrossRef](#)]

129. Sachindra, D.; Huang, F.; Barton, A.; Perera, B. Least square support vector and multi-linear regression for statistically downscaling general circulation model outputs to catchment streamflows. *Int. J. Climatol.* **2013**, *33*, 1087–1106. [[CrossRef](#)]
130. De'ath, G.; Fabricius, K.E. Classification and regression trees: A powerful yet simple technique for ecological data analysis. *Ecology* **2000**, *81*, 3178–3192. [[CrossRef](#)]
131. Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Spatial prediction of flood susceptible areas using rule based decision tree (DT) and a novel ensemble bivariate and multivariate statistical models in GIS. *J. Hydrol.* **2013**, *504*, 69–79. [[CrossRef](#)]
132. Dehgani, M.; Saghafian, B.; Rivaz, F.; Khodadadi, A. Evaluation of dynamic regression and artificial neural networks models for real-time hydrological drought forecasting. *Arabian J. Geosci.* **2017**, *10*, 266. [[CrossRef](#)]
133. Choubin, B.; Zehtabian, G.; Azareh, A.; Rafiee-Sardooi, E.; Sajedi-Hosseini, F.; Kişi, Ö. Precipitation forecasting using classification and regression trees (CART) model: A comparative study of different approaches. *Environ. Earth Sci.* **2018**, *77*, 314. [[CrossRef](#)]
134. Choubin, B.; Darabi, H.; Rahmati, O.; Sajedi-Hosseini, F.; Kløve, B. River suspended sediment modelling using the cart model: A comparative study of machine learning techniques. *Sci. Total Environ.* **2018**, *615*, 272–281. [[CrossRef](#)] [[PubMed](#)]
135. Liaw, A.; Wiener, M. Classification and regression by randomforest. *R News* **2002**, *2*, 18–22.
136. Wang, Z.; Lai, C.; Chen, X.; Yang, B.; Zhao, S.; Bai, X. Flood hazard risk assessment model based on random forest. *J. Hydrol.* **2015**, *527*, 1130–1141. [[CrossRef](#)]
137. Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in GIS. *J. Hydrol.* **2014**, *512*, 332–343. [[CrossRef](#)]
138. Bui, D.T.; Tuan, T.A.; Klempe, H.; Pradhan, B.; Revhaug, I. Spatial prediction models for shallow landslide hazards: A comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree. *Landslides* **2016**, *13*, 361–378.
139. Etemad-Shahidi, A.; Mahjoobi, J. Comparison between m5' model tree and neural networks for prediction of significant wave height in lake superior. *Ocean Eng.* **2009**, *36*, 1175–1181. [[CrossRef](#)]
140. Dietterich, T.G. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
141. Sajedi-Hosseini, F.; Malekian, A.; Choubin, B.; Rahmati, O.; Cipullo, S.; Coulon, F.; Pradhan, B. A novel machine learning-based approach for the risk assessment of nitrate groundwater contamination. *Sci. Total Environ.* **2018**, *644*, 954–962. [[CrossRef](#)]
142. Moore, K.J.; Kurt, M.; Eriten, M.; McFarland, D.M.; Bergman, L.A.; Vakakis, A.F. Wavelet-bounded empirical mode decomposition for measured time series analysis. *Mech. Syst. Signal Process.* **2018**, *99*, 14–29. [[CrossRef](#)]
143. Wang, W.-C.; Chau, K.-W.; Xu, D.-M.; Chen, X.-Y. Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. *Water Resour. Manag.* **2015**, *29*, 2655–2675. [[CrossRef](#)]
144. Al-Musaylh, M.S.; Deo, R.C.; Li, Y.; Adamowski, J.F. Two-phase particle swarm optimized-support vector regression hybrid model integrated with improved empirical mode decomposition with adaptive noise for multiple-horizon electricity demand forecasting. *Appl. Energy* **2018**, *217*, 422–439. [[CrossRef](#)]
145. Ouyang, Q.; Lu, W.; Xin, X.; Zhang, Y.; Cheng, W.; Yu, T. Monthly rainfall forecasting using EEMD-SVR based on phase-space reconstruction. *Water Resour. Manag.* **2016**, *30*, 2311–2325. [[CrossRef](#)]
146. Zhang, J.; Hou, G.; Ma, B.; Hua, W. Operating characteristic information extraction of flood discharge structure based on complete ensemble empirical mode decomposition with adaptive noise and permutation entropy. *J. Vib. Control* **2018**. [[CrossRef](#)]
147. Badrzadeh, H.; Sarukkalige, R.; Jayawardena, A. Hourly runoff forecasting for flood risk management: Application of various computational intelligence models. *J. Hydrol.* **2015**, *529*, 1633–1643. [[CrossRef](#)]
148. Kim, G.; Barros, A.P. Quantitative flood forecasting using multisensor data and neural networks. *J. Hydrol.* **2001**, *246*, 45–62. [[CrossRef](#)]
149. Saghafian, B.; Haghnegahdar, A.; Dehgani, M. Effect of ENSO on annual maximum floods and volume over threshold in the southwestern region of Iran. *Hydrol. Sci. J.* **2017**, *62*, 1039–1049. [[CrossRef](#)]
150. Kourgialas, N.N.; Dokou, Z.; Karatzas, G.P. Statistical analysis and ann modeling for predicting hydrological extremes under climate change scenarios: The example of a small Mediterranean Agro-watershed. *J. Environ. Manag.* **2015**, *154*, 86–101. [[CrossRef](#)] [[PubMed](#)]

151. Panda, R.K.; Pramanik, N.; Bala, B. Simulation of river stage using artificial neural network and mike 11 hydrodynamic model. *Comput. Geosci.* **2010**, *36*, 735–745. [[CrossRef](#)]
152. Noori, R.; Karbassi, A.; Farokhnia, A.; Dehghani, M. Predicting the longitudinal dispersion coefficient using support vector machine and adaptive neuro-fuzzy inference system techniques. *Environ. Eng. Sci.* **2009**, *26*, 1503–1510. [[CrossRef](#)]
153. Pereira Filho, A.J.; dos Santos, C.C. Modeling a densely urbanized watershed with an artificial neural network, weather radar and telemetric data. *J. Hydrol.* **2006**, *317*, 31–48. [[CrossRef](#)]
154. Jingyi, Z.; Hall, M.J. Regional flood frequency analysis for the Gan-Ming River basin in China. *J. Hydrol.* **2004**, *296*, 98–117. [[CrossRef](#)]
155. Ahmad, S.; Simonovic, S.P. An artificial neural network model for generating hydrograph from hydro-meteorological parameters. *J. Hydrol.* **2005**, *315*, 236–251. [[CrossRef](#)]
156. Ju, Q.; Yu, Z.; Hao, Z.; Ou, G.; Zhao, J.; Liu, D. Division-based rainfall-runoff simulations with BP neural networks and Xinanjiang model. *Neurocomputing* **2009**, *72*, 2873–2883. [[CrossRef](#)]
157. Sahoo, G.B.; Ray, C.; De Carlo, E.H. Use of neural network to predict flash flood and attendant water qualities of a mountainous stream on Oahu, Hawaii. *J. Hydrol.* **2006**, *327*, 525–538. [[CrossRef](#)]
158. Ghose, D.K. *Measuring Discharge Using Back-Propagation Neural Network: A Case Study on Brahmani River Basin*; Springer: Singapore, 2018; pp. 591–598.
159. Pan, H.-X.; Cheng, G.-J.; Cai, L. Comparison of the extreme learning machine with the support vector machine for reservoir permeability prediction. *Comput. Eng. Sci.* **2010**, *2*, 37.
160. Chang, F.-J.; Chen, P.-A.; Lu, Y.-R.; Huang, E.; Chang, K.-Y. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* **2014**, *517*, 836–846. [[CrossRef](#)]
161. Shen, H.-Y.; Chang, L.-C. Online multistep-ahead inundation depth forecasts by recurrent NARX networks. *Hydrol. Earth Syst. Sci.* **2013**, *17*, 935–945. [[CrossRef](#)]
162. Bruen, M.; Yang, J. Functional networks in real-time flood forecasting—A novel application. *Adv. Water Resour.* **2005**, *28*, 899–909. [[CrossRef](#)]
163. Chiang, Y.-M.; Chang, F.-J.; Jou, B.J.-D.; Lin, P.-F. Dynamic ANN for precipitation estimation and forecasting from radar observations. *J. Hydrol.* **2007**, *334*, 250–261. [[CrossRef](#)]
164. Bhattacharya, B.; Solomatine, D.P. Neural networks and M5 model trees in modelling water level-discharge relationship. *Neurocomputing* **2005**, *63*, 381–396. [[CrossRef](#)]
165. Heiser, M.; Scheidl, C.; Eisl, J.; Spangl, B.; Hübl, J. Process type identification in torrential catchments in the eastern Alps. *Geomorphology* **2015**, *232*, 239–247. [[CrossRef](#)]
166. Khosravi, K.; Pham, B.T.; Chapi, K.; Shirzadi, A.; Shahabi, H.; Revhaug, I.; Prakash, I.; Bui, D.T. A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at haraz watershed, Northern Iran. *Sci. Total Environ.* **2018**, *627*, 744–755. [[CrossRef](#)] [[PubMed](#)]
167. Aichouri, I.; Hani, A.; Bougherira, N.; Djabri, L.; Chaffai, H.; Lallaheem, S. River flow model using artificial neural networks. *Energy Procedia* **2015**, *74*, 1007–1014. [[CrossRef](#)]
168. Torabi, M.; Hashemi, S.; Saybani, M.R.; Shamshirband, S.; Mosavi, A. A Hybrid clustering and classification technique for forecasting short-term energy consumption. *Environ. Prog. Sustain. Energy* **2018**, *47*. [[CrossRef](#)]
169. Adamowski, J.F. Development of a short-term river flood forecasting method for snowmelt driven floods based on wavelet and cross-wavelet analysis. *J. Hydrol.* **2008**, *353*, 247–266. [[CrossRef](#)]
170. Leahy, P.; Kiely, G.; Corcoran, G. Structural optimisation and input selection of an artificial neural network for river level prediction. *J. Hydrol.* **2008**, *355*, 192–201. [[CrossRef](#)]
171. Wei, C.C. Soft computing techniques in ensemble precipitation nowcast. *Appl. Soft Comput. J.* **2013**, *13*, 793–805. [[CrossRef](#)]
172. R Schiffer, R.A.; Rossow, W.B. The International Satellite Cloud Climatology Project (ISCCP): The first project of the world climate research programme. *Bull. Am. Meteorol. Soc.* **1983**, *64*, 779–784. [[CrossRef](#)]
173. Castillo, E. Functional networks. *Neural Process. Lett.* **1998**, *7*, 151–159. [[CrossRef](#)]
174. Jimeno-Sáez, P.; Senent-Aparicio, J.; Pérez-Sánchez, J.; Pulido-Velazquez, D.; Cecilia, J.M. Estimation of instantaneous peak flow using machine-learning models and empirical formula in peninsular Spain. *Water* **2017**, *9*, 347.
175. Chang, F.-J.; Chang, Y.-T. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Adv. Water Resour.* **2006**, *29*, 1–10. [[CrossRef](#)]

176. Mosavi, A.; Lopez, A.; Varkonyi-Koczy, A.R. Industrial Applications of Big Data: State of the Art Survey. In *Recent Advances in Technology Research and Education*; Springer: Cham, Switzerland, 2017; pp. 225–232.
177. Rezaeianzadeh, M.; Tabari, H.; Yazdi, A.A.; Isik, S.; Kalin, L. Flood flow forecasting using ANN, ANFIS and regression models. *Neural Comput. Appl.* **2014**, *25*, 25–37. [[CrossRef](#)]
178. Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Flood susceptibility analysis and its verification using a novel ensemble support vector machine and frequency ratio method. *Stoch. Environ. Res. Risk Assess.* **2015**, *29*, 1149–1165. [[CrossRef](#)]
179. Young, C.C.; Liu, W.C.; Wu, M.C. A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events. *Appl. Soft Comput. J.* **2017**, *53*, 205–216. [[CrossRef](#)]
180. Altunkaynak, A.; Nigussie, T.A. Prediction of daily rainfall by a hybrid wavelet-season-neuro technique. *J. Hydrol.* **2015**, *529*, 287–301. [[CrossRef](#)]
181. Chang, L.-C.; Shen, H.-Y.; Chang, F.-J. Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. *J. Hydrol.* **2014**, *519*, 476–489. [[CrossRef](#)]
182. Nanda, T.; Sahoo, B.; Beria, H.; Chatterjee, C. A wavelet-based non-linear autoregressive with exogenous inputs (WNARX) dynamic neural network model for real-time flood forecasting using satellite-based rainfall products. *J. Hydrol.* **2016**, *539*, 57–73. [[CrossRef](#)]
183. French, J.; Mawdsley, R.; Fujiyama, T.; Achuthan, K. Combining machine learning with computational hydrodynamics for prediction of tidal surge inundation at estuarine ports. *Procedia IUTAM* **2017**, *25*, 28–35. [[CrossRef](#)]
184. Hong, W.-C. Rainfall forecasting by technological machine learning models. *Appl. Math. Comput.* **2008**, *200*, 41–57. [[CrossRef](#)]
185. Pan, T.-Y.; Yang, Y.-T.; Kuo, H.-C.; Tan, Y.-C.; Lai, J.-S.; Chang, T.-J.; Lee, C.-S.; Hsu, K.H. Improvement of watershed flood forecasting by typhoon rainfall climate model with an ANN-based southwest monsoon rainfall enhancement. *J. Hydrol.* **2013**, *506*, 90–100. [[CrossRef](#)]
186. Rajurkar, M.; Kothiyari, U.; Chaube, U. Modeling of the daily rainfall-runoff relationship with artificial neural network. *J. Hydrol.* **2004**, *285*, 96–113. [[CrossRef](#)]
187. Hsu, M.-H.; Lin, S.-H.; Fu, J.-C.; Chung, S.-F.; Chen, A.S. Longitudinal stage profiles forecasting in rivers for flash floods. *J. Hydrol.* **2010**, *388*, 426–437. [[CrossRef](#)]
188. Pang, B.; Guo, S.; Xiong, L.; Li, C. A nonlinear perturbation model based on artificial neural network. *J. Hydrol.* **2007**, *333*, 504–516. [[CrossRef](#)]
189. Doycheva, K.; Horn, G.; Koch, C.; Schumann, A.; König, M. Assessment and weighting of meteorological ensemble forecast members based on supervised machine learning with application to runoff simulations and flood warning. *Adv. Eng. Inform.* **2017**, *33*, 427–439. [[CrossRef](#)]
190. Fleming, S.W.; Bourdin, D.R.; Campbell, D.; Stull, R.B.; Gardner, T. Development and operational testing of a super-ensemble artificial intelligence flood-forecast model for a pacific northwest river. *J. Am. Water Resour. Assoc.* **2015**, *51*, 502–512. [[CrossRef](#)]
191. Choubin, B.; Khalighi, S.S.; Malekian, A. *Impacts of Large-Scale Climate Signals on Seasonal Rainfall in the Maharlou-Bakhtegan Watershed*; Journal of Range and Watershed Management: Kashan, Iran, 2016.
192. Kisi, O.; Sanikhani, H. Prediction of long-term monthly precipitation using several soft computing methods without climatic data. *Int. J. Climatol.* **2015**, *35*, 4139–4150. [[CrossRef](#)]
193. Liang, Z.; Li, Y.; Hu, Y.; Li, B.; Wang, J. A data-driven SVR model for long-term runoff prediction and uncertainty analysis based on the Bayesian framework. *Theor. Appl. Climatol.* **2018**, *133*, 137–149. [[CrossRef](#)]
194. Han, S.; Coulibaly, P. Bayesian flood forecasting methods: A review. *J. Hydrol.* **2017**, *551*, 340–351. [[CrossRef](#)]
195. Choubin, B.; Moradi, E.; Golshan, M.; Adamowski, J.; Sajedi-Hosseini, F.; Mosavi, A. An Ensemble prediction of flood susceptibility using multivariate discriminant analysis, classification and regression trees, and support vector machines. *Elsevier Sci. Total Environ.* **2018**, *651*, 2087–2096. [[CrossRef](#)] [[PubMed](#)]
196. Teng, J.; Jakeman, A.; Vaze, J.; Croke, B.F.; Dutta, D.; Kim, S. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environ. Model. Softw.* **2017**, *90*, 201–216. [[CrossRef](#)]
197. Elsafi, S.H. Artificial neural networks (ANNs) for flood forecasting at Dongola station in the river Nile, Sudan. *Alex. Eng. J.* **2014**, *53*, 655–662. [[CrossRef](#)]
198. Mohammadzadeh, D.; Bazaz, J.B.; Yazd, S.V.J.; Alavi, A.H. Deriving an intelligent model for soil compression index utilizing multi-gene genetic programming. *Springer Environ. Earth Sci.* **2016**, *75*, 262. [[CrossRef](#)]

199. Singh, P.; Borah, B. Indian summer monsoon rainfall prediction using artificial neural network. *Stoch. Environ. Res. Risk Assess.* **2013**, *27*, 1585–1599. [[CrossRef](#)]
200. Shamim, M.A.; Hassan, M.; Ahmad, S.; Zeeshan, M. A comparison of artificial neural networks (ANN) and local linear regression (LLR) techniques for predicting monthly reservoir levels. *KSCE J. Civ. Eng.* **2016**, *20*, 971–977. [[CrossRef](#)]
201. Rezaeian-Zadeh, M.; Tabari, H.; Abghari, H. Prediction of monthly discharge volume by different artificial neural network algorithms in semi-arid regions. *Arabian J. Geosci.* **2013**, *6*, 2529–2537. [[CrossRef](#)]
202. Gazendam, E.; Gharabaghi, B.; Ackerman, J.D.; Whiteley, H. Integrative neural networks models for stream assessment in restoration projects. *J. Hydrol.* **2016**, *536*, 339–350. [[CrossRef](#)]
203. Lin, J.-Y.; Cheng, C.-T.; Chau, K.-W. Using support vector machines for long-term discharge prediction. *Hydrol. Sci. J.* **2006**, *51*, 599–612. [[CrossRef](#)]
204. Cunningham, S.C.; Griffioen, P.; White, M.D.; Nally, R.M. Assessment of ecosystems: A system for rigorous and rapid mapping of floodplain forest condition for Australia’s most important river. *Land Degrad. Dev.* **2018**, *29*, 127–137. [[CrossRef](#)]
205. Ahmad, S.; Kalra, A.; Stephen, H. Estimating soil moisture using remote sensing data: A machine learning approach. *Adv. Water Resour.* **2010**, *33*, 69–80. [[CrossRef](#)]
206. Prasad, R.D.; Ravinesh, C.; Li, Y.; Maraseni, T. Input selection and performance optimization of ANN-based streamflow forecasts in the drought-prone murray darling basin region using IIS and MODWT algorithm. *Atmos. Res.* **2017**, *197*, 42–63. [[CrossRef](#)]
207. Cannas, B.; Fanni, A.; Sias, G.; Tronci, S.; Zedda, M.K. River flow forecasting using neural networks and wavelet analysis. *Geophys. Res. Abstr.* **2005**, *7*, 08651.
208. Najafi, B. An Intelligent Artificial Neural Network-Response Surface Methodology Method for Accessing the Optimum Biodiesel and Diesel Fuel Blending Conditions in a Diesel Engine from the Viewpoint of Exergy and Energy Analysis. *Energies* **2018**, *11*, 860. [[CrossRef](#)]
209. Singh, R.M. Wavelet-ANN model for flood events. In Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011), Patiala, India, 20–22 December 2011; pp. 165–175.
210. Ramana, R.V.; Krishna, B.; Kumar, S.R.; Pandey, N.G. Monthly rainfall prediction using wavelet neural network analysis. *Water Resour. Manag.* **2013**, *27*, 3697–3711. [[CrossRef](#)]
211. Tantanee, S.; Patamatammakul, S.; Oki, T.; Sriboonlue, V.; Prempree, T. Coupled wavelet-autoregressive model for annual rainfall prediction. *J. Environ. Hydrol.* **2005**, *13*, 124–146.
212. Mekanik, F.; Imteaz, M.A.; Talei, A. Seasonal rainfall forecasting by adaptive network-based fuzzy inference system (ANFIS) using large scale climate signals. *Clim. Dyn.* **2016**, *46*, 3097–3111. [[CrossRef](#)]
213. Wang, W.C.; Chau, K.W.; Cheng, C.T.; Qiu, L. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *J. Hydrol.* **2009**, *374*, 294–306. [[CrossRef](#)]
214. Kisi, O.; Nia, A.M.; Gosheh, M.G.; Tajabadi, M.R.J.; Ahmadi, A. Intermittent streamflow forecasting by using several data driven techniques. *Water Resour. Manag.* **2012**, *26*, 457–474. [[CrossRef](#)]
215. Li, C.; Guo, S.; Zhang, J. Modified NLPM-ANN model and its application. *J. Hydrol.* **2009**, *378*, 137–141. [[CrossRef](#)]
216. Huang, S.; Chang, J.; Huang, Q.; Chen, Y. Monthly streamflow prediction using modified EMD-based support vector machine. *J. Hydrol.* **2014**, *511*, 764–775. [[CrossRef](#)]
217. Zhu, S.; Zhou, J.; Ye, L.; Meng, C. Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze River, China. *Environ. Earth Sci.* **2016**, *75*, 531. [[CrossRef](#)]
218. Bass, B.; Biedent, P. Surrogate modeling of joint flood risk across coastal watersheds. *J. Hydrol.* **2018**, *558*, 159–173. [[CrossRef](#)]
219. Araghinejad, S.; Azmi, M.; Khalghi, M. Application of artificial neural network ensembles in probabilistic hydrological forecasting. *J. Hydrol.* **2011**, *407*, 94–104. [[CrossRef](#)]
220. Tan, Q.-F.; Lei, X.-H.; Wang, X.; Wang, H.; Wen, X.; Ji, Y.; Kang, A.-Q. An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. *J. Hydrol.* **2018**. [[CrossRef](#)]
221. Nosratabadi, S.; Mosavi, A. Sustainable Business Model: A Review. *Preprints* **2018**, *2018100378*. [[CrossRef](#)]
222. Høverstad, B.A.; Tidemann, A.; Langseth, H.; Öztürk, P. Short-term load forecasting with seasonal decomposition using evolution for parameter tuning. *IEEE Trans. Smart Grid* **2015**, *6*, 1904–1913. [[CrossRef](#)]

223. Tantithamthavorn, C.; McIntosh, S.; Hassan, A.E.; Matsumoto, K. Automated parameter optimization of classification techniques for defect prediction models. In Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), Austin, TX, USA, 14–22 May 2016; pp. 321–332.
224. Varkonyi-Kóczy, A.R. Review on the usage of the multiobjective optimization package of modefrontier in the energy sector. In *Recent Advances in Technology Research and Education*; Springer: Cham, Switzerland, 2017; p. 217.
225. Dineva, A.; Várkonyi-Kóczy, A.R.; Tar, J.K. Anytime fuzzy supervisory system for signal auto-healing. In *Advanced Materials Research*; Trans Tech Publications: Tihany, Hungary, 2015; pp. 269–272.
226. Torabi, M.; Mosavi, A.; Ozturk, P.; Varkonyi-Kóczy, A.; Istvan, V. A hybrid machine learning approach for daily prediction of solar radiation. In *Recent Advances in Technology Research and Education*; Springer: Cham, Switzerland, 2018; pp. 266–274.
227. Solgi, A.; Nourani, V.; Pourhaghi, A. Forecasting daily precipitation using hybrid model of wavelet-artificial neural network and comparison with adaptive neurofuzzy inference system (case study: Verayneh station, Nahavand). *Adv. Civ. Eng.* **2014**, *2014*, 279368. [[CrossRef](#)]
228. Badrzadeh, H.; Sarukkalige, R.; Jayawardena, A. Improving ann-based short-term and long-term seasonal river flow forecasting with signal processing techniques. *River Res. Appl.* **2016**, *32*, 245–256. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Forward Prediction of Runoff Data in Data-Scarce Basins with an Improved Ensemble Empirical Mode Decomposition (EEMD) Model

Yinghao Yu <sup>1</sup>, Hongbo Zhang <sup>1,2,\*</sup> and Vijay P. Singh <sup>3</sup>

<sup>1</sup> School of Environmental Science and Engineering, Chang'an University, Xi'an 710054, China; 2015229036@chd.edu.cn

<sup>2</sup> Key Laboratory of Subsurface Hydrology and Ecological Effect in Arid Region, Ministry of Education, Chang'an University, Xi'an 710054, China

<sup>3</sup> Department of Biological and Agricultural Engineering & Zachry Department of Civil Engineering, Texas A&M University, College Station, TX 77843, USA; vsingh@tamu.edu

\* Correspondence: hbzhang@chd.edu.cn; Tel.: +86-029-8233-9959

Received: 24 February 2018; Accepted: 23 March 2018; Published: 27 March 2018

**Abstract:** Data scarcity is a common problem in hydrological calculations that often makes water resources planning and engineering design challenging. Combining ensemble empirical mode decomposition (EEMD), a radial basis function (RBF) neural network, and an autoregression (AR) model, an improved EEMD prediction model is proposed for runoff series forward prediction, i.e., runoff series extension. In the improved model, considering the decomposition-prediction-reconstruction principle, EEMD was employed for decomposition and reconstruction and the RBF and AR model were used for component prediction. Also, the method of tracking energy differences (MTED) was used as stopping criteria for EEMD in order to solve the problem of mode mixing that occurs frequently in EEMD. The orthogonality index (Ort) and the relative average deviation (RAD) were introduced to verify the mode mixing and prediction performance. A case study showed that the MTED-based decomposition was significantly better than decomposition methods using the standard deviation (SD) criteria and the G. Rilling (GR) criteria. After MTED-based decomposition, mode mixing in EEMD was suppressed effectively ( $|Ort| < 0.23$ ) and stable orthogonal components were obtained. For this, annual runoff series forward predictions using the improved EEMD-based prediction model were significantly better ( $RAD < 11.1\%$ ) than predictions by the rainfall-runoff method and the AR model method. Thus, this forward prediction model can be regarded as an approach for hydrological series extension, and shows promise for practical applications.

**Keywords:** data scarce basins; runoff series; data forward prediction; ensemble empirical mode decomposition (EEMD); stopping criteria; method of tracking energy differences (MTED)

## 1. Introduction

Hydrological data scarcity is a constant challenge for international hydrological research. In 2003, the International Association of Hydrological Sciences (IAHS) launched an initiative called “predictions in ungauged basins (PUB)” for the IAHS Decade at the 23rd International Union of Geodesy and Geophysics (IUGG) in Sapporo, Japan. This initiative strongly promoted the development of hydrological research in ungauged basins [1]. In 2013, a new science decade of IAHS was approved, “Panta Rhei—Everything Flows”, which made global hydrological researchers aware of the slow progress in developing innovative hydrological research methods to solve the problem of hydrological data scarcity [2]. It is well known that runoff data are the most important hydrological data for river-basin management and are fundamental to hydraulic engineering design and water-resource

management. If a catchment has few or no runoff data, then it is difficult to carry out policies and strategies in water-resource management. Therefore, it is important to develop innovative methods to address the problem of runoff data scarcity to service global hydrological research and engineering design.

To address the scarcity of runoff data, many researchers have proposed various prediction methods. They are generally divided into two major categories: physics-based and data-driven methods. The physics-based methods usually build a proper hydrological model in the few- or no-data catchments according to the catchment condition, and obtain some unknown model parameters directly from other river basins that have observed data. When meteorological data and underlying surface data in the few- or no-data catchments are available, the hydrological processes can be simulated to obtain the runoff data series in the ungauged basins or extend the series in the few-data basins. In recent years, many physics-based methods have been proposed to undertake the predictions in ungauged basins. Servat et al. [3] developed two rainfall-runoff models (GR3 and CREC) which can do runoff prediction from ungauged basins on the basis of land use and rainfall distribution over the year. McIntyre et al. [4] proposed a new approach to the regionalization of conceptual rainfall-runoff models based on ensemble modeling and model averaging. Model parameters were calibrated for 10 gauged basins with hydrological conditions similar to those of the ungauged basins. Also, ensemble predictions of runoff were done for ungauged basins. Wan et al. [5] developed a lumped conceptual rainfall-runoff model for rapid runoff prediction in south Florida with a unique and complicated hydrological setting. Li et al. [6] evaluated two regionalization approaches, spatial proximity and physical similarity, by which two runoff models (SIMHYD and GR4J) were used to predict runoff from the Yarlung Tsangpo River basin. Because these models are based on physical causes, the procedure is very complex and highly susceptible to factors such as the integrity and accuracy of the data on the river basin's underlying surface conditions, spatial-temporal variance of meteorological data, complexity of rainfall-runoff process, and limited understanding of circulation patterns of water in the basins [7]. In recent years, the precision of predictions by hydrological simulations has been found to be far from satisfactory in some regions, so hybrid models coupling physics-based models with data-driven methods have gained more attention.

Data-driven methods are generally used to make short-term predictions or data extension using mathematical methods and intelligent algorithms via the statistical characteristics of short observational runoff series or unknown meteorological and hydrological black-box models in data-scarce basins or reference watersheds. Besaw et al. [8] developed and tested two artificial neural networks (ANNs) to predict runoff from the Winooski River basin with time-lagged records of precipitation and temperature as input data. Mohamoud [9] employed flow duration curves for forecasting flow in ungauged basins by combining dominant landscape and climate descriptors from 29 nearby catchments with multiple regression. It is well-known that data-driven methods require less data and have a simpler structure than physics-based methods. Furthermore, data-driven methods have a good prediction performance without really simulating the rainfall-runoff process, and can avoid the complex physical process and the influence of model uncertainty. Thus, data-driven methods are usually used as alternative and similar or even superior to those of physics-based methods in ungauged basins where hydrological model simulations cannot be carried out effectively. Nowadays, they have been widely used in hydraulic engineering design. However, they are not universal and are affected by regional conditions. For instance, in north-western China, due to the poor similarity of reference basins and the complicated and changeable rainfall-runoff relationships in the region, rainfall data in the basin and the hydrological characteristics in the reference basin cannot be used as data-driven model inputs in the region. Therefore, the prediction of runoff series in such regions should preferably be based on the existing short runoff data than the unsatisfied rainfall-runoff model or poor similarity of the reference basin. Generally, data extension based on the existing runoff data is called forward prediction, which means predicting a non-measured runoff process before the existing runoff records by data-extension methods.

In the steady state, time-series models and artificial intelligence algorithms, such as artificial neural networks (ANNs) [10–14] and support vector machines (SVMs) [15–18], can make satisfactory predictions. However, under the dual influence of global climate change and intense human activities in recent years, runoff series have exhibited such characteristics as high complexity, non-stationarity, non-linearity and multiple time-scales [19–21]. These characteristics make analysis of runoff characteristics and conventional hydrological time-series forecasting more difficult. The precision of conventional prediction methods does not satisfy the requirements of current engineering design and hydrological research. Therefore, a new prediction method should be developed for hydrological data extension to meet hydraulic engineering design demands.

The multiple time scales of hydrological time series refer to the existence of multi-level time scales and local features in the hydrological series changes in the time domain. For multiple time-scales issues with non-stationarity and non-linearity variables, many time-scale decomposition approaches have been introduced to separate the different time scales in hydrological series for hydrological prediction and to provide important support for system analysis and runoff prediction. For example, the wavelet transform (WT) has been adopted by many researchers for analyzing hydrological time series with multiple scales due to its excellence in situations with multiple resolutions in time and frequency domains [22–25]. Essentially, a wavelet transform is a Fourier transform with an adjustable window, and the signal should be stable in the WT window. Therefore, it is still susceptible to the limitations of Fourier analysis. Although WT provides high resolution in both the frequency domain and the time domain, certain limitations of this method may generate some false harmonic waves. Thus, the selection of WT basis functions is critical and has a significant impact on the wavelet decomposition performance. In order to promote the development of multiple time-scale analysis approaches, Huang proposed a novel signal analysis method in 1998 called empirical mode decomposition (EMD) [26]. This method is essentially the smoothing treatment of the signal, by which the multi-scale fluctuation or trend components in the signal are decomposed to generate a series of intrinsic mode functions (IMFs) and a residual. Comparing two approaches, it can be seen that an EMD-based Hilbert spectrum and a wavelet spectrum have the same characteristics on the linear framework, while the Hilbert spectrum has significantly higher resolution in both time and frequency domains. Therefore, it is often considered that the EMD result can reflect non-stationary and non-linearity characteristics in the original series more accurately than the WT method, and EMD is regarded as a more effective way to process complex signals. In classical hydrology, a hydrological time series can be regarded as a set of random components, periodical components and trend component. When the decomposition result of the EMD is perfect, the high-frequency components, the low-frequency components and the residual obtained by the decomposition can be approximated as random components, periodic components, and the trend [27,28]. Nowadays, EMD has become a new method for multi-time-scale analysis of non-stationary hydrological time series and has been successfully applied in hydrological research around the world [29,30]. Based on the EMD method, researchers have proposed “decomposition-prediction-reconstruction” coupling models which improve the precision of hydrological prediction effectively [31–33]. However, limitations still exist, such as mode mixing and IMFs’ orthogonality effect on the EMD performance and prediction precision. Ideally, each component obtained after the decomposition should contain information on one time scale. However, due to the defects of the decomposition method and the random fluctuations in hydrological series, a component obtained after decomposition may contain different information belonging to other components. That is called mode mixing, which will lead to an unclear physical meaning of each component and confusion in further analysis. The orthogonality of EMD can be understood mathematically in that each IMF decomposed is orthogonal and also can be understood in the decomposing operation in that there is no energy loss of the original series in the process of extracting components in the ideal state. Unfortunately, the total energy of the components is always significantly different from the energy of the original series in the actual EMD process. To address these issues, Wu and Huang proposed the ensemble EMD (EEMD) method to suppress mode mixing

in EMD [34,35]. However, EEMD is not perfect, and mode mixing still occurs among low-frequency components. So, it is inferred that a proper stopping criteria is critical for EMD, which can guarantee that the EEMD method can provide satisfactory decomposition results [36]. For hydrological time series, a proper stopping criteria not only improves the precision of fluctuation component extraction in hydrological time-series data, but also preserves the long-range trend in the series to the greatest extent possible, which significantly influences the accuracy of forward prediction. In the EEMD, the stopping criteria is called the SD criteria, in which the standard deviation is used to stop the decomposition procedure [26]. Later, G. Rilling proposed the so-called G. Rilling (GR) criteria [37], which leverages the evaluation function  $\sigma(t)$  and the predefined threshold to control when the sifting process stops. In the GR criteria, two conditions are to be fulfilled: the number of extrema and the number of zero-crossings must differ at most by 1, and the mean between the upper and lower envelopes must be close to zero. Compared with the SD criteria proposed by Huang, the GR can obtain the mean value of IMFs more accurately. However, the effects of these criteria are still limited, and the energy loss during sifting and orthogonality among IMFs are not fully addressed. Some problems such as mode mixing cannot still be solved perfectly. To address these issues, Cheng proposed a new EMD sifting stopping criteria, the method of tracking energy differences (MTED), aiming to solve the mode-mixing problem from the perspective of energy. Currently, it has achieved excellent results in fault diagnosis [38].

In view of the above analysis, this paper investigated the applicability of the sifting stopping criteria to hydrological time series. The MTED was selected as the sifting stopping criterion for decomposing runoff series using EEMD. For forward prediction, a radial basis function (RBF) neural network, and an autoregressive (AR) model were combined to create a “decomposition-prediction-reconstruction”-based improved EEMD prediction model in order to predict short runoff series and further solve the problem of runoff data scarcity encountered in hydrological research and engineering design.

## 2. Materials and Methods

### 2.1. Empirical Mode Decomposition (EMD)

EMD is a new and innovative self-adaptive time-frequency signal-processing method proposed by Huang in 1998 [29]. This method is primarily designed for non-stationary and non-linear data. Signal decomposition obtains multiple stable IMFs and a monotonic residual based on the data's own time-scale pattern. In hydrological applications, EMD converts a non-stationary hydrological series into a series of hydrological components with clear patterns that have specific physical meanings [33]. These components are more predictable and can improve the precision of forward prediction significantly. Details of the EMD procedure are as follows:

- Step 1: Identify all local maxima and minima in the original time series  $X(t)$ . The upper and lower envelopes of the time series are obtained by cubic spline interpolation. The mean of the upper and lower enveloping lines is  $m(t)$ :

$$m(t) = \frac{X_{\max}(t) + X_{\min}(t)}{2} \quad (1)$$

- Step 2: A new series  $h(t)$  is calculated by subtracting the mean  $m(t)$  from the original series  $X(t)$ :

$$h(t) = X(t) - m(t) \quad (2)$$

- Step 3: The EMD sifting stopping criteria determines whether sifting should stop. If the stopping condition is met,  $h(t)$  is the IMF, and the next step is executed. If the stopping condition is not met, then  $h(t)$  is used as the original series, steps 1 and 2 are repeated until the stopping condition is met, and the first IMF, IMF1  $c_1(t)$ , is calculated.

- Step 4: The residual series  $r_1(t)$  is obtained by subtracting the IMF  $c_1(t)$  from the original series  $X(t)$ :

$$r_1(t) = X(t) - c_1(t) \quad (3)$$

- Step 5: The residual series  $r_1(t)$  is used as the new original series, and steps 1–4 are repeated. All the IMFs,  $c_1(t)$ ,  $c_2(t)$ , ...,  $c_n(t)$ , are decomposed until  $c_n(t)$  is a monotonic or single-extreme-point residual.

## 2.2. EEMD

The EEMD method is an improvement of EMD method that reduces mode mixing and obtains the actual time-frequency distribution of the signal [35]. The principle is to leverage the statistical features (uniform frequency distribution) of Gaussian white noise. When white noise is added to a signal, the signal becomes continuous on different scales to reduce mode mixing. Details of the decomposition principle and procedure are as follows:

- Step 1: White noise  $n_i(t)$  with a mean of 0 and standard deviation constant is added to the original signal  $X(t)$  multiple times. The standard deviation of the white noise is set to 0.1–0.4 times the standard deviation of the original signal (0.2 in this study):

$$X_i(t) = X(t) + n_i(t) \quad (4)$$

where  $X_i(t)$  represents the signal after the  $i$ -th addition of Gaussian white noise.

- Step 2: Each  $X_i(t)$  undergoes the EMD procedure. The IMF component obtained is denoted by  $c_{ij}(t)$ , and the residual term is denoted by  $r_i(t)$ . Among them,  $c_{ij}(t)$  represents the  $j$ -th IMF from the decomposition of the signal after the  $i$ -th addition of Gaussian white noise.
- Step 3: Steps 1 and 2 are repeated  $N$  times. Based on the principle that the statistical mean of an uncorrelated random series is 0, the IMFs are subjected to an overall averaging operation to eliminate the impact of adding Gaussian white noise to the actual IMF multiple times. Finally, the IMF obtained from EEMD is as follows:

$$c_j(t) = \frac{1}{N} \sum_{i=1}^N c_{ij} \quad (5)$$

where  $c_j(t)$  represents the  $j$ -th IMF of the original signal obtained by EEMD. As the value of  $N$  increases, the sum of IMFs for the corresponding white noise approaches 0. At this point, the result of EEMD is as follows:

$$X(t) = \sum_j c_j(t) + r(t) \quad (6)$$

where  $r(t)$  is the final residual, which represents the average trend of the signal. Any signal  $X(t)$  can be decomposed into multiple IMFs and one residual via EEMD. IMF  $c_j(t)$  ( $j = 1, 2, \dots$ ) represents the signal's components from high frequency to low frequency. Each frequency contains distinct components and varies with the signal  $X(t)$ .

## 2.3. Improved Ensemble Empirical Mode Decomposition (EEMD)

Whether the decomposed IMFs are proper or applicable is largely determined by the sifting stopping criteria. Different criteria result in different IMFs from decomposition. Due to the limited applicability of the SD criteria proposed by Huang [34,35] and the GR criteria proposed by G. Rilling [37], the method of tracking energy differences (MTED) is introduced as the sifting stopping criteria to improve the EEMD method.

The MTED is different from the other two sifting stopping criteria. It assumes that the IMFs are finite and orthogonal to each other; that is, in an ideal state, when an IMF is sifted, no energy is lost

during sifting. If the EEMD exhibits a smaller energy loss during sifting, it is more likely to guarantee the orthogonality of IMFs and, therefore, the EEMD sifting is more appropriate. It is clear that the MTED mainly works from the perspective of energy and ensures that each extracted IMF and residual are orthogonal in terms of energy. Details of the procedure are as follows [38]:

$$E_X = \int_{-\infty}^{\infty} \left[ \sum_{i=1}^n c_i(t) \right]^2 dt = \int_{-\infty}^{\infty} c_1^2(t) dt + \int_{-\infty}^{\infty} c_2^2(t) dt + \cdots + \int_{-\infty}^{\infty} c_n^2(t) dt = E_1 + E_2 + \cdots + E_n \quad (7)$$

where  $E_X$  is the total energy of the series;  $c_i(t)$  is an IMF or residual of the original series after EEMD; and  $E_1, E_2, \dots, E_n$  is the energy of the corresponding component.

During EEMD, if a component  $h(t) = X(t) - m(t)$  is obtained, then when  $h(t)$  is sifted from  $X(t)$ , the sum of the energy for  $h(t)$  and the rest of the series is as follows:

$$E_{\text{tot}} = \int_{-\infty}^{\infty} h^2(t) dt + \int_{-\infty}^{\infty} m^2(t) dt = E_h + E_m \quad (8)$$

Then, the difference between the total series energy before and after  $h(t)$  is sifted is as follows:

$$E_{\text{err}} = E_{\text{tot}} - E_X \quad (9)$$

Normally,  $|E_{\text{err}}|$  decreases as the number of sifting increases. If after the  $k$ -th sifting,  $|E_{\text{err}}|$  is greater than that it was after the  $(k-1)$ -th sifting, then it is considered that  $|E_{\text{err}}|$  has reached its minimum after the  $(k-1)$ -th sifting, no more sifting is needed, and this round of sifting stops. The  $h(t)$  obtained from the  $(k-1)$ -th sifting is selected as an IMF, and the next step of the EEMD is executed to obtain other IMFs and the residual. If the condition is not met, then sifting is repeated until an IMF is obtained.

#### 2.4. Improved EEMD-Based Decomposition-Prediction-Reconstruction Model

To forward predict or extend short observational runoff series in data-scarce catchments, an improved EEMD-based method and the prediction model are combined in this paper to create an improved EEMD prediction model according to the “decomposition-prediction-reconstruction” principle. As follows from the previous analysis, the low-frequency components and residual terms of runoff time series calculated using EEMD have regular and stable fluctuation. Thus, an AR model prediction can provide high precision. In comparison, the high-frequency component (IMF1) has significant fluctuations and strong non-linearity and the AR model prediction designed for a stable series is hardly satisfactory. Therefore, in this study, a RBF neural network, which is suitable for processing non-linear series, was employed for prediction. Moreover, it was also discovered that IMF1 components from original runoff series obtained by EEMD demonstrated fluctuations and variations consistent with rainfall series in the same basin. To avoid the problem that runoff predictions have only statistical significance instead of physical representations, the rainfall series in the same period was used in this paper as one of the input vectors for the RBF neural network. Additionally, since runoff series exhibit strong auto-correlation and this may still exist in IMF1, a partial autocorrelation function (PACF) and the Akaike information criteria (AIC) [39] were employed for autocorrelation analysis and to select the inputs of the RBF neural network (the strongest three orders as additional input vectors).

In general, the procedure for the improved EEMD prediction model is summarized as follows: a short observed runoff series undergoes orthogonal decomposition via the improved EEMD method to obtain several IMFs and one residual; i.e., a non-stationary runoff time series is decomposed into multiple quasi-stable components and one trend component. Then, IMF1 undergoes forward prediction using the RBF neural network, and the terms from IMF2 to the residual undergo forward

prediction via the AR model. The forward predicted components are reconstructed to obtain runoff data for years when measured data are missing. After verification, the obtained runoff series are combined with the original series to generate a runoff series that meets length requirements for water resources engineering design or hydrological research.

### 3. Results and Discussion

#### 3.1. Case Selection

In some remote regions of north-western China, hydrological stations are scarce, and the length of hydrological data series is seriously insufficient. With poor rainfall-runoff relations, data scarcity has become a major issue for hydraulic engineering design and hydrological research development in the region. Zhaoshiyao hydrological station in the Wuding River basin and the Suide hydrological station in the Dali River (a tributary of the Wuding River) basin in north-western China are typical of data-scarce stations. Thus, they were selected as research stations in this paper. Annual runoff data series (1971–2010) at two hydrological stations were selected as the study subject, in which annual runoff data in 1981–2010 (30 years) were used as training data, and annual runoff data in 1971–1980 (10 years) were used as verification data. The data were collected from hydrological manuals published by the Hydrological Bureau of the Yellow River Conservancy Commission (YRCC).

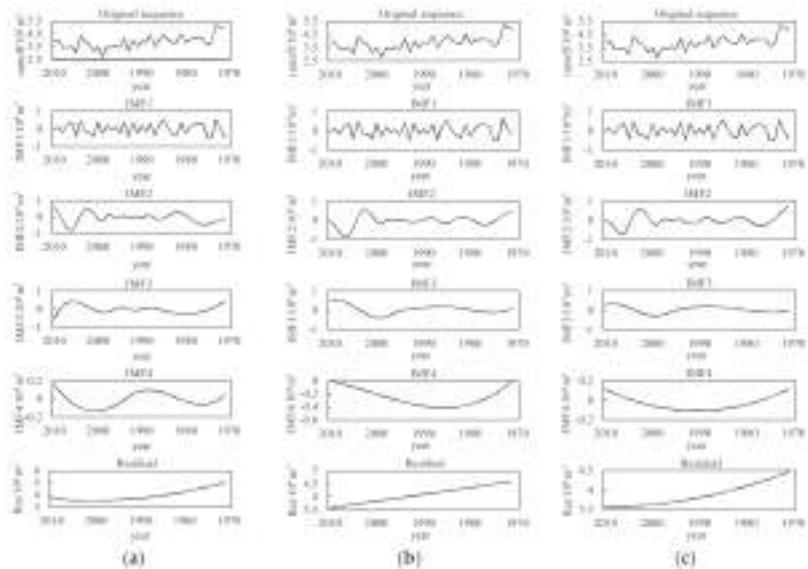
#### 3.2. Calculation and Analysis

##### 3.2.1. Improved EEMD

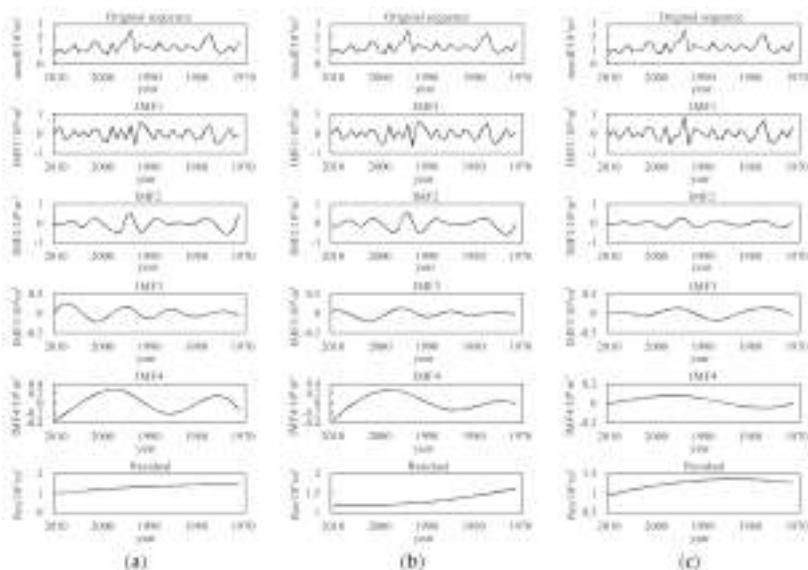
The annual runoff data in 1971–2010 at two hydrological stations were reversed and decomposed using the improved EEMD method. The MTED was used as the stopping criteria for EEMD sifting. To verify the decomposition performance by the improved EEMD method, the results were compared with the results obtained by decomposition methods based on the SD criteria and the GR criteria, as shown in Figures 1 and 2.

In Figure 1a,b and Figure 2a,b show SD and GR criteria-based EEMD components, respectively, and Figures 1c and 2c shows the MTED-based components. These figures show that with the three sifting stopping criteria, four IMFs and one residual can be obtained from decomposition. However, the same original series were decomposed into different components (different IMFs and different residuals) based on the three sifting stopping criteria. When the SD criteria or the GR criteria was used as the EEMD sifting stopping criteria, the decomposed components were highly fluctuating and irregular. In particular, in the low frequency component (such as IMF4), the decomposed components exhibited irregular waveforms. In other words, SD and GR criteria-based components exhibited severe mode mixing such that this did not accurately show hydrological fluctuations or periodical changes. Such fluctuating and irregular components were difficult to predict due to their weak regularity. In contrast, the components obtained by the MTED were relatively stable, fluctuating around 0, and had regular waveforms. After several rounds of sifting, the low-frequency components demonstrated regular sinusoidal fluctuations. This means that the EEMD results obtained by the MTED were better because mode mixing in the process was suppressed effectively, and the decomposed IMF component was more stable, which provided a solid foundation for forward prediction in the next stage. It is worth mentioning that the extraction of each component except IMF1 is based on the previous extracted component in the decomposition process of EMD. Different sifting stopping criteria could make extraction different, and the difference will enlarge along with the decomposition. Although the difference is not obvious among the high-frequency components (such as IMF1 and IMF2) obtained under three criteria, it is an objective reality and would lead to the curves of the low-frequency components being obviously different, as shown in Figures 1 and 2. Compared with the MTED, the low-frequency component obtained through SD- and GR-criteria show irregular fluctuations,

which indicates that the MTED-based EMD performs better than SD and GR criteria-based EMD in separating the multi-time scale information from the original series.



**Figure 1.** The decomposition results of runoff series based on (a) standard deviation (SD) criteria; (b) G. Rilling (GR) criteria; and (c) the method of tracking energy differences (MTED) at the Zhaoshiyao station.



**Figure 2.** The decomposition results of runoff series based on (a) standard deviation (SD) criteria; (b) G. Rilling (GR) criteria; and (c) the method of tracking energy differences (MTED) at the Suide station.

To further verify the above statement, the orthogonality index  $\text{Ort}$  was used to evaluate the superiority of the three sifting stopping criteria.  $\text{Ort}$  is an index that evaluates the orthogonality of the IMF components, and its value closer to zero means that the IMF components are more orthogonal [40]. The principle is illustrated as follows [41]:

Original runoff data undergo EEMD, and then  $n$  IMF components and one residual are obtained. The corresponding formula is as follows:

$$X(t) = \sum_{q=1}^n c_q(t) + r(t) \quad (10)$$

where  $c_q(t)$  is the  $q$ -th IMF and  $r(t)$  is the residual which is defined as the last IMF component, i.e.,  $r(t)$  is defined as  $c_{n+1}(t)$ . Then, the original runoff data are represented as follows:

$$X(t) = \sum_{q=1}^{n+1} c_q(t) \quad (11)$$

The runoff data  $X(t)$  in the form of a square are as follows:

$$X^2(t) = \sum_{q=1}^{n+1} c_q^2(t) + 2 \sum_{q=1}^{n+1} \sum_{p=1}^{n+1} c_q(t)c_p(t) (q \neq p) \quad (12)$$

If all the IMF components are orthogonal, then the second term on the right side of the equal sign in above formula should be zero. Therefore, the definition of the orthogonality index  $\text{Ort}$  is as follows:

$$\text{Ort} = \sum_{t=1}^N \left( \frac{\sum_{p=1}^{n+1} \sum_{q=1}^{n+1} c_p(t)c_q(t)}{X^2(t)} \right) (p \neq q) \quad (13)$$

where  $N$  is the length of the runoff series.

Next, all IMFs and residuals decomposed with the three sifting stopping criteria are taken through the Hilbert–Huang transform (HHT) to obtain their Hilbert spectrum [26]. After integrating the Hilbert spectrum with respect to time, their Hilbert marginal spectrum is obtained, respectively, as shown in Figures 3 and 4, in which it is seen that the Hilbert spectrum accurately reflects variations in the component's amplitude with time and frequency. The marginal spectrum statistically represents the accumulated amplitude distribution of each component along the frequency. The orthogonality of the components is represented by the coincidence of major frequencies in the marginal spectrum. A smaller coincidence means the orthogonality is superior.

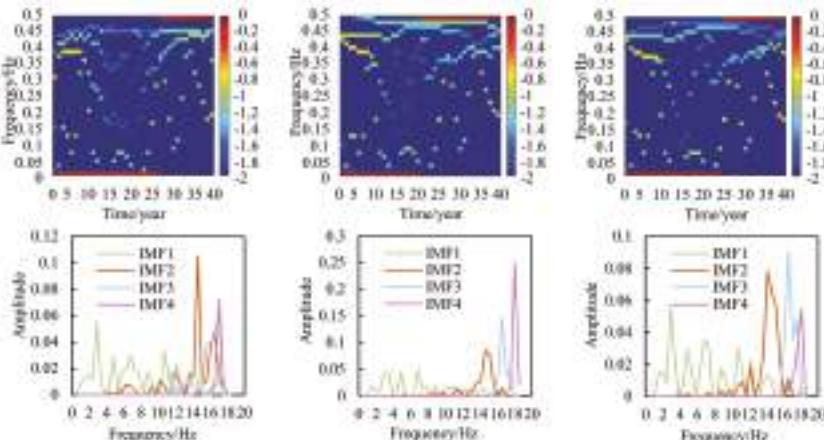
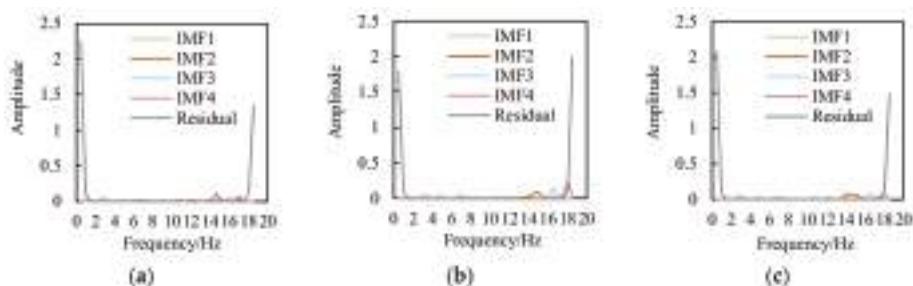
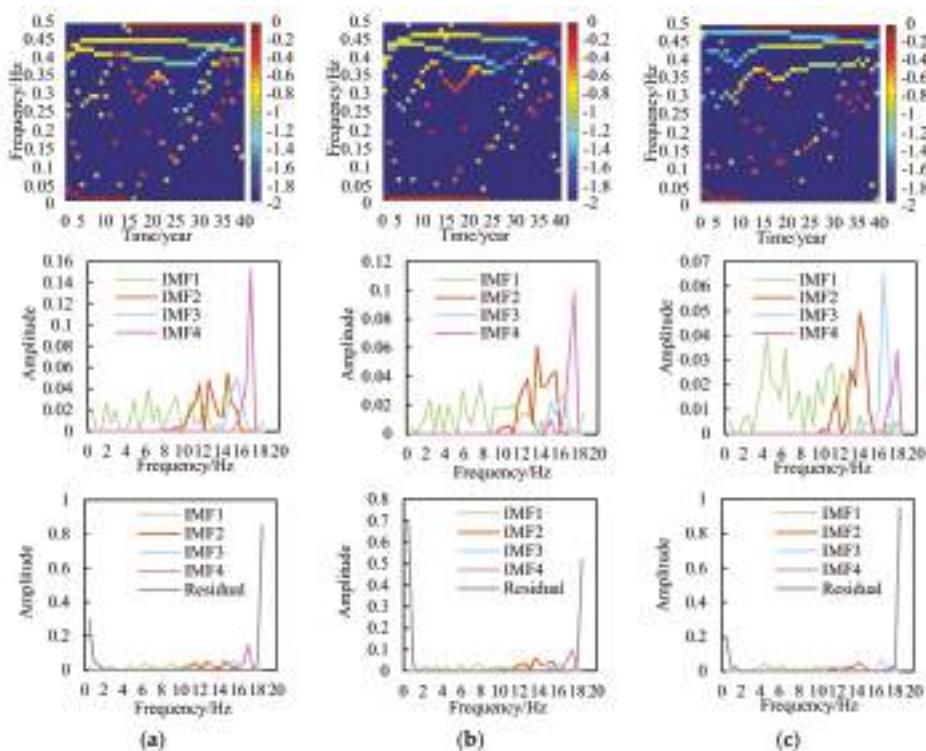


Figure 3. Cont.



**Figure 3.** Hilbert spectrums and marginal spectra of runoff components obtained based on (a) SD criteria; (b) GR criteria; and (c) MTED at the Zhaoshiyao station.



**Figure 4.** Hilbert spectrums and marginal spectra of runoff components obtained based on (a) SD criteria; (b) GR criteria; and (c) MTED at the Suide station.

Table 1 lists the orthogonality indexes of the EEMD results with three sifting stopping criteria at the two hydrological stations. “With the residual” means the residual is used as the last IMF component in the calculation, and “without the residual” represents the fact that the residual has been removed, and other IMFs are used in the calculation. It can be found whether or not the residual is taken into the calculation, the MTED-based orthogonality index is closer to 0 than the indexes by the SD criteria and the GR criteria. This indicates that, compared with the SD criteria-based or GR criteria-based decomposition components, the MTED-based components are more orthogonal or with less mode mixing. This statement is also supported to a certain extent in Figures 3 and 4. In the Hilbert

spectrum of two figures, the horizontal axis represents time, the vertical axis represents frequency, and the depth of the color describes the magnitude of the amplitude. Although the Hilbert spectrums of three stopping criteria do not show a significant difference, it can be seen that the spectrum of the MTED-based components are more recognizable and regular than SD criteria-based and GR criteria-based components. Furthermore, a more significant superiority of MTED to SD criteria and GR criteria can be seen in the marginal spectrum, in which the dominant frequency and frequency band of each component can be recognized well. As shown in the marginal spectrum of Figures 3 and 4, the dominant frequency of MTED-based components is significant while that of SD criteria-based and GR criteria-based components is difficult to distinguish. The detailed representation is as follows: the frequency band of MTED-based components is distributed relatively independently on different frequencies while that of SD criteria-based and GR criteria-based components overlap in the frequency range. In other words, Figures 3 and 4 show that MTED-based decomposition is superior to SD criteria-based and GR criteria-based decomposition in decomposing the original series into several components corresponding to different frequency bands (different time scales). All these indicate that the SD criteria-based and GR criteria-based decomposition components have serious mode mixing and poor orthogonality. Fortunately, the MTED-based improved EEMD method can suppress mode mixing in the EEMD effectively, generating stable IMFs representing multi-scale physical information and thereby illustrating hydrological periodical change hidden in the runoff data to the extent possible.

**Table 1.** Orthogonality index of runoff components based on three stopping criteria.

Orthogonality Index	Zhaoshiyao Station			Suide Station		
	SD Criteria	GR Criteria	MTED	SD Criteria	GR Criteria	MTED
Without residual	−0.10	−0.10	−0.08	−1.24	−0.95	−0.23
With residual	−1.72	−3.57	−0.66	−8.63	−9.86	−5.64

### 3.2.2. Radial Basis Function (RBF) Neural Network and Autoregression (AR) Model Prediction

The data of runoff components (1981–2010) decomposed by the improved EEMD were used as training data. Through training, a RBF neural network and AR model was built and used for the prediction, in which the RBF neural network was employed to forward predict or extend the IMF1 data during non-observed period (1971–1980) by coupling the rainfall data in the same period, and the AR model was used to forward predict or extend other components' data (IMF2–4 and the residual) during the non-observed period. Next, all the predicted runoff components were combined to obtain the predicted annual runoff data for the non-observed period.

To verify the prediction effect and compare the impact of three sifting stopping criteria (the SD criteria, the GR criteria and the MTED) on runoff prediction, the prediction results by the EEMD prediction models with three criteria were compared, as listed in Table 2. Here, measured runoff data in the verification period were used as the benchmark for error analysis, and the relative average deviation (RAD) and the Nash–Sutcliffe efficiency (NSE) were used as error evaluation indexes to undertake comprehensive measurement and evaluation of the prediction performance. A smaller RAD and a larger NSE represent higher prediction precision.

Table 2 shows the prediction performance of EEMD prediction models based on the SD criteria, the GR criteria and the MTED, where it is clear that the MTED-based EEMD prediction model has significantly more precision than the SD-based and GR-based models. This also supports the inference about the applicability and superiority of MTED as a stopping criteria for EMD sifting.

**Table 2.** Error assessment of improved EEMD prediction models based on three stopping criteria.

Error Evaluation Index	Zhaoshiyao Station			Suide Station		
	SD Criteria	GR Criteria	MTED	SD Criteria	GR Criteria	MTED
Relative average deviation (RAD)/%	9.45	9.39	6.86	25.24	25.60	11.10
Nash-Sutcliffe efficiency (NSE)	0.19	-0.24	0.40	0.34	0.29	0.89

### 3.3. Result Verification

To further verify the prediction performance of the improved EEMD prediction model, two common forward prediction methods used in engineering design (the rainfall-runoff method and the AR model) were selected for comparison, and measured runoff data during 1971–1980 were used for verification. The rainfall-runoff method is primarily based on the rainfall-runoff correlation in the research basin. Rainfall data was measured data in the study basin, and hence the missing annual runoff data can be predicted by the rainfall-runoff regression equation established in the training period (1981–2010). In the AR model, the measured runoff data from 1981 to 2010 were first sorted in reverse time order. After determining the three most significant orders as model inputs, the reverse data were implemented in the AR procedure to estimate the missing runoff data from 1980 to 1971. Table 3 lists the results of evaluation by error in the prediction by the three forward prediction methods. It shows that compared with the conventional rainfall-runoff method and the AR model method, the improved EEMD prediction model had more precise prediction.

**Table 3.** Error assessment of forward prediction by three models.

Error Evaluation Index	Zhaoshiyao Station			Suide Station		
	AR Model Method	Rainfall-Runoff Method	Improved EEMD Prediction Model	AR Model Method	Rainfall-Runoff Method	Improved EEMD Prediction Model
RAD/%	11.03	15.13	6.86	27.76	19.67	11.10
NSE	-0.87	-1.54	0.40	-0.02	-0.11	0.89

To test the usefulness of these three forward prediction methods in engineering design, 40-year (1971–2010) runoff data series were generated, including 10-year predicted runoff data and 30-year measured data, and these were then compared with 40-year measured runoff data via statistical parameters. The results are shown in Table 4.

The table shows that the extended long runoff series by the improved EEMD model had similar statistical parameters with the measured runoff data. If the designer used the extended data by the improved EEMD prediction model for engineering design, he would get a better hydrological design value to meet the engineering design requirements than by using the other two methods. However, the designer is likely to result in design deviation and put the project at risk if he adopted the extended data by the other two methods in order to undertake engineering design. Therefore, the improved EEMD model is undoubtedly a better choice for engineering design and hydrological research when hydrological data is scarce in a basin or region similar to north-west China, by which the obtained design value has a significant advantage for the regional water resource supply-demand balance and the safety of hydraulic project operation.

**Table 4.** Statistical parameters between extended runoff series by different models and observed series at the Zhaoshiyao and Suide stations.

Statistical Parameters	Zhaoshiyao Station			Suide Station		
	Mean	Mean Square Error	Coefficient of Variation	Mean	Mean Square Error	Coefficient of Variation
Original sequence	3.86	0.51	0.13	1.28	0.40	0.31
Improved EEMD prediction model	3.84	0.44	0.11	1.29	0.39	0.30
Rainfall-runoff method	3.70	0.38	0.10	1.22	0.33	0.27
AR model	3.75	0.37	0.10	1.26	0.33	0.26

#### 4. Conclusions

In this paper a new method, called the improved EEMD prediction model, is proposed to forward predict or extend runoff series in data-scarce basins for serving regional hydraulic engineering design. The model combines ensemble empirical mode decomposition (EEMD), a radial basis function (RBF) neural network, and an auto-regression (AR) model, whereby the EEMD is employed for decomposition and reconstruction, and the RBF and AR model are employed for forward predicting or extending the IMFs and residual components. Also, three EMD sifting stopping criteria (the SD criteria, the GR criteria and the MTED) are discussed and compared in this study to find the best criteria to solve the problem of mode mixing and improve the decomposition quality of EEMD. Additionally, two quantitative evaluation measures, the relative average deviation (RAD) and the Nash–Sutcliffe efficiency (NSE), are used to evaluate the performance of the improved prediction model and compare them with the AR model and a rainfall-runoff method.

The case study at two hydrological gauges located in north-west China, the Zhaoshiyao and Suide stations, indicates that: (1) the improved EEMD using the MTED as sifting stopping criteria suppresses mode mixing effectively ( $|Ort| < 0.23$ ), ensuring that the IMFs are quasi-stable to preserve the physical information and periodical change contained in the runoff data to the extent possible; (2) the improved EEMD prediction model has lower RAD and NSE statistics, 6.86% and 0.40 at the Zhaoshiyao station, respectively, and 11.10% and 0.89 at the Suide station, respectively, and these are significantly better than the rainfall-runoff method and the AR model.

Comparative results indicate that this forward prediction model is undoubtedly a better choice for engineering design and hydrological research when hydrological data is scarce in a basin or region similar to north-west China.

**Acknowledgments:** The research was supported by the National Natural Science Foundation of China (51379014), the Major Program of the National Natural Science Foundation of China (41790441) and the Technology Foundation for Selected Overseas Chinese Scholars, Department of Personnel in Shaanxi Province of China (2017035). The rainfall data was obtained from China Meteorological Data Service Center (<http://data.cma.cn/en>). The authors would like to thank the reviewers for their insightful comments that greatly improved the quality of the paper.

**Author Contributions:** H.Z. provided the writing ideas and supervised the study; Y.Y. conceived and designed the methods; Y.Y. and V.P.S. wrote the paper, and all the authors were responsible for data processing and data analysis.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Sivapalan, M.; Takeuchi, K.; Franks, S.W.; Gupta, V.K.; Karambiri, H.; Lakshmi, V.; Liang, X.; McDonnell, J.J.; Mendiondo, E.M.; O'Connell, P.E. Iahs decade on predictions in ungauged basins (pub), 2003–2012: Shaping an exciting future for the hydrological sciences. *Int. Assoc. Sci. Hydrol. Bull.* **2003**, *48*, 857–880. [[CrossRef](#)]
2. Montanari, A.; Young, G.; Savenije, H.H.G.; Hughes, D.; Wagener, T.; Ren, L.L.; Koutsoyiannis, D.; Cudennec, C.; Toth, E.; Grimaldi, S. “Panta rhei-everything flows”: Change in hydrology and society—the iahs scientific decade 2013–2022. *Int. Assoc. Sci. Hydrol. Bull.* **2013**, *58*, 1256–1275. [[CrossRef](#)]
3. Servat, E.; Dezetter, A. Rainfall-runoff modelling and water resources assessment in northwestern ivory coast. Tentative extension to ungauged catchments. *J. Hydrol.* **1993**, *148*, 231–248. [[CrossRef](#)]
4. McIntyre, N.; Lee, H.; Wheater, H.; Young, A.; Wagener, T. Ensemble predictions of runoff in ungauged catchments. *Water Resour. Res.* **2005**, *41*, 4203–4206. [[CrossRef](#)]
5. Wan, Y.; Konyha, K. A simple hydrologic model for rapid prediction of runoff from ungauged coastal catchments. *J. Hydrol.* **2015**, *528*, 571–583. [[CrossRef](#)]
6. Li, F.; Zhang, Y.; Xu, Z.; Liu, C.; Zhou, Y.; Liu, W. Runoff predictions in ungauged catchments in southeast tibetan plateau. *J. Hydrol.* **2014**, *511*, 28–38. [[CrossRef](#)]
7. Liu, C.; Bai, P.; Wang, Z.; Liu, S.; Liu, X. Study on prediction of ungauged basins: A case study on the tibetan plateau. *J. Hydraul. Eng.* **2016**, *47*, 272–282.

8. Besaw, L.E.; Rizzo, D.M.; Bierman, P.R.; Hackett, W.R. Advances in ungauged streamflow prediction using artificial neural networks. *J. Hydrol.* **2010**, *386*, 27–37. [[CrossRef](#)]
9. Mohamoud, Y. Prediction of daily flow duration curves and streamflow for ungauged catchments using regional flow duration curves. *Int. Assoc. Sci. Hydrol. Bull.* **2008**, *53*, 706–724. [[CrossRef](#)]
10. Gazzaz, N.M.; Yusoff, M.K.; Aris, A.Z.; Juahir, H.; Ramli, M.F. Artificial neural network modeling of the water quality index for kinta river (malaysia) using water quality variables as predictors. *Mar. Pollut. Bull.* **2012**, *64*, 2409–2420. [[CrossRef](#)] [[PubMed](#)]
11. Kalin, L.; Isik, S.; Schoonover, J.E.; Lockaby, B.G. Predicting water quality in unmonitored watersheds using artificial neural networks. *J. Environ. Qual.* **2010**, *39*, 1429–1440. [[CrossRef](#)] [[PubMed](#)]
12. Noori, N.; Kalin, L. Coupling swat and ann models for enhanced daily streamflow prediction. *J. Hydrol.* **2016**, *533*, 141–151. [[CrossRef](#)]
13. Palani, S.; Tkalich, P.; Balasubramanian, R.; Palanichamy, J. Ann application for prediction of atmospheric nitrogen deposition to aquatic ecosystems. *Mar. Pollut. Bull.* **2011**, *62*, 1198–1206. [[CrossRef](#)] [[PubMed](#)]
14. Sahoo, G.B.; Ray, C.; Carlo, E.H.D. Use of neural network to predict flash flood and attendant water qualities of a mountainous stream on oahu, hawaii. *J. Hydrol.* **2006**, *327*, 525–538. [[CrossRef](#)]
15. Asefa, T.; Kemblowski, M.; McKee, M.; Khalil, A. Multi-time scale stream flow predictions: The support vector machines approach. *J. Hydrol.* **2006**, *318*, 7–16. [[CrossRef](#)]
16. Jajarmizadeh, M.; Lafdani, E.K.; Harun, S.; Ahmadi, A. Application of svm and swat models for monthly streamflow prediction, a case study in south of iran. *KSCE J. Civ. Eng.* **2015**, *19*, 345–357. [[CrossRef](#)]
17. Kalra, A.; Ahmad, S. Using oceanic-atmospheric oscillations for long lead time streamflow forecasting. *Water Resour. Res.* **2009**, *45*, 450–455. [[CrossRef](#)]
18. Noori, R.; Karbassi, A.R.; Moghaddamnia, A.; Han, D.; Zokaei-Ashtiani, M.H.; Farokhnia, A.; Gousheh, M.G. Assessment of input variables determination on the svm model performance using pca, gamma test, and forward selection techniques for monthly stream flow prediction. *J. Hydrol.* **2011**, *401*, 177–189. [[CrossRef](#)]
19. Manuca, R.; Savit, R. Stationarity and nonstationarity in time series analysis. *Phys. D Nonlinear Phenom.* **1996**, *99*, 134–161. [[CrossRef](#)]
20. Trenberth, K.E. Recent observed interdecadal climate changes in the northern hemisphere. *Bull. Am. Meteorol. Soc.* **1990**, *71*, 377–390. [[CrossRef](#)]
21. Tsonis, A.A. Widespread increases in low-frequency variability of precipitation over the past century. *Nature* **1996**, *382*, 700–702. [[CrossRef](#)]
22. Kumar, P.; Foufoula-Georgiou, E. A multicomponent decomposition of spatial rainfall fields: 1. Segregation of large- and small-scale features using wavelet transforms. *Water Resour. Res.* **1993**, *29*, 2515–2532. [[CrossRef](#)]
23. Saco, P.; Kumar, P. Coherent modes in multiscale variability of streamflow over the united states. *Water Resour. Res.* **2000**, *36*, 1049–1068. [[CrossRef](#)]
24. Sang, Y.F.; Wang, Z.; Liu, C. Discrete wavelet-based trend identification in hydrologic time series. *Hydrolog. Process.* **2013**, *27*, 2021–2031. [[CrossRef](#)]
25. Wang, W.; Jing, D. Wavelet network model and its application to the prediction of hydrology. *Nat. Sci.* **2003**, *1*, 67–71.
26. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.C.; Chi, C.T.; Liu, H.H. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
27. Sang, Y.F.; Wang, Z.; Liu, C. Comparison of the mk test and emd method for trend identification in hydrological time series. *J. Hydrol.* **2014**, *510*, 293–298. [[CrossRef](#)]
28. Sang, Y.F.; Wang, Z.; Liu, C. Period identification in hydrologic time series using empirical mode decomposition and maximum entropy spectral analysis. *J. Hydrol.* **2012**, *s 424–425*, 154–164. [[CrossRef](#)]
29. Huang, Y.; Schmitt, F.G.; Lu, Z.; Liu, Y. Analysis of daily river flow fluctuations using empirical mode decomposition and arbitrary order hilbert spectral analysis. *J. Hydrol.* **2009**, *373*, 103–111. [[CrossRef](#)]
30. Li, X.; Ding, Z. Emd method for multiple time-scale analysis on fluctuation characteristic of natural annual runoff time series of fen river. *Water Resour. Power* **2008**, *30*–32.
31. Huang, S.; Chang, J.; Huang, Q.; Chen, Y. Monthly streamflow prediction using modified emd-based support vector machine. *J. Hydrol.* **2014**, *511*, 764–775. [[CrossRef](#)]
32. Karthikeyan, L.; Kumar, D.N. Predictability of nonstationary time series using wavelet and emd based arma models. *J. Hydrol.* **2013**, *502*, 103–119. [[CrossRef](#)]

33. Zhang, H.; Singh, V.P.; Wang, B.; Yu, Y. Ceref: A hybrid data-driven model for forecasting annual streamflow from a socio-hydrological system. *J. Hydrol.* **2016**, *540*, 246–256. [[CrossRef](#)]
34. Huang, N.E. A study of the characteristics of white noise using the empirical mode decomposition method. *Proc. Math. Phys. Eng. Sci.* **2004**, *460*, 1597–1611.
35. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2005**, *1*, 1–41. [[CrossRef](#)]
36. Huang, N.E.; Shen, Z.; Long, S.R. A new view of nonlinear water waves: The hilbert spectrum1. *Ann. Rev. Fluid Mech.* **1999**, *31*, 417–457. [[CrossRef](#)]
37. Gabriel, R.; Patrick, E.; Paulo, G. On empirical mode decomposition and its algorithms. In Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP-03), Grado, Italy, 8–11 June 2003.
38. Cheng, J. *Research on Fault Diagnosis Methods for Rotating Machinery Based on Hilbert-Huang Transform*; Hunan University: Changsha, China, 2005.
39. Pan, W. Akaike’s information criteria in generalized estimating equations. *Biometrics* **2001**, *57*, 120–125. [[CrossRef](#)] [[PubMed](#)]
40. Ahmed, N.; Rao, K.R.; Debnath, L. Orthogonal transforms for digital signal processing. *IEEE Trans. Syst. Man Cybern.* **1976**, *9*, 66–67. [[CrossRef](#)]
41. Li, X. *Study on Orthogonality of Emd Method in Hht*; Kunming University Of Science And Technology: Kunming, China, 2010.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning

Yang Chen <sup>1,2,\*</sup>, Rongshuang Fan <sup>2</sup>, Xiucheng Yang <sup>3</sup>, Jingxue Wang <sup>1</sup> and Aamir Latif <sup>4</sup>

<sup>1</sup> School of Geomatics, Liaoning Technical University, Fuxin 123000, China; xiaoxue1861@163.com

<sup>2</sup> Chinese Academy of Surveying and Mapping, Beijing 100830, China; fanrsh@casm.ac.cn

<sup>3</sup> ICube Laboratory, University of Strasbourg, 67000 Strasbourg, France; xiuchengyang@163.com

<sup>4</sup> Institute of Geographic Sciences and Natural Resources Research, University of Chinese Academy of Sciences, Beijing 10010, China; aamir.latif@igsnrr.ac.cn

\* Correspondence: chenyang1017@126.com; Tel.: +86-153-5020-5816

Received: 21 March 2018; Accepted: 23 April 2018; Published: 1 May 2018

**Abstract:** Accurate information on urban surface water is important for assessing the role it plays in urban ecosystem services in the context of human survival and climate change. The precise extraction of urban water bodies from images is of great significance for urban planning and socioeconomic development. In this paper, a novel deep-learning architecture is proposed for the extraction of urban water bodies from high-resolution remote sensing (HRRS) imagery. First, an adaptive simple linear iterative clustering algorithm is applied for segmentation of the remote-sensing image into high-quality superpixels. Then, a new convolutional neural network (CNN) architecture is designed that can extract useful high-level features of water bodies from input data in a complex urban background and mark the superpixel as one of two classes: an including water or no-water pixel. Finally, a high-resolution image of water-extracted superpixels is generated. Experimental results show that the proposed method achieved higher accuracy for water extraction from the high-resolution remote-sensing images than traditional approaches, and the average overall accuracy is 99.14%.

**Keywords:** deep learning; convolutional neural networks; superpixel; urban water bodies; high-resolution remote-sensing images

## 1. Introduction

Urban water bodies are important parts of the urban ecosystem that are of great significance for urban environmental testing, urban heat-island effects, and urban ecosystem maintenance [1]. The changes in urban water bodies make a huge difference to human lives and may cause disasters, such as surface subsidence, urban inland inundation and health problems [2]. Therefore, it is necessary to know about urban water distribution and changes in the water area.

In recent years, satellite remote-sensing technology has developed rapidly and has the characteristics of a wide observation range, short return period, and so on [3]. It has been widely used in many fields such as military reconnaissance, environmental protection, mapping and geography [4]. Among current urban water-extraction technologies, a mainstream method uses remote-sensing imagery to gather urban water information in a timely and accurate way [5]. Previous urban water-resource surveys have been based on low- and medium-resolution images [6]. However, small water bodies such as small ponds and narrow rivers cannot be extracted due to the limited spatial resolution of these remote-sensing images. With the improvement of the spatial resolution of remote-sensing images, many remote-sensing satellites (such as Gaofeng-2, Ziyuan-3, WorldView-2,

IKONOS and RapidEye) can provide high-resolution images. Most high-resolution remote-sensing images only have four bands (blue, green, red and near-infrared), lacking the SWIR necessary to compute the modified normalized difference water index (MNDWI) and the automated water extraction index (AWEI) indices [7]. A high-resolution spatial multi-spectral image has more detailed spatial features information, which can greatly improve the accuracy of urban water body extraction [8].

Many algorithms have been proposed for identifying water bodies with remote-sensing imagery including single-band threshold and multi-band threshold methods, water body index methods, sub-pixel water mapping methods, and supervised and unsupervised classification methods [9,10]. The water body index method has the characteristics of fast calculation and high precision, so it is widely used in practical applications. McFeeters proposed the normalized difference water index (NDWI) model and the basic idea of this model is based on a normalized difference vegetation index (NDVI) [11]. However, this model is unable to distinguish between dark shadow and water bodies [12]. Xu proposed the MNDWI which uses mid-infrared bands for normalization instead of near-infrared and green bands, and has better results for urban water body extraction [13]. These improvements in the water index are generally difficult to apply in high-resolution remote-sensing images due to limited spectral resolution. Image classification methods such as supervised or machine learning are often used to extract water bodies from remote-sensing images [14]. Generally, machine-learning methods include neural network and support vector machine, and unsupervised classification methods include k-means clustering and ISODATA clustering methods [15,16]. The above algorithms are mainly used on low spatial resolution remote-sensing images. The existing algorithms have undergone less research for urban water body extraction in high-resolution satellite images. At present, the main problem for extracting an urban water body by low spectral resolution remote-sensing images is the ability to distinguish between the building shadows and the water bodies which is one of the most difficult tasks [17].

Deep learning is the learning process that simulates the human brain [18]. It can automatically extract high-level features from low-level features of the input image [19,20]. In this study, a novel method for the extraction of urban water bodies based on deep learning is proposed for high spatial resolution multi-spectral images. A new convolutional neural network (CNN) architecture is designed that can extract water and detect building shadows effectively even in complex circumstances and predict the superpixel as one of two classes including water and no water.

The major contributions of this paper are:

- (1) A novel extraction method for urban water bodies based on deep learning is proposed for remote-sensing images. The proposed method combines the superpixel method with deep learning to extract urban water bodies and distinguish shadow from water.
- (2) A new CNN architecture is designed, which can learn the characteristics of water bodies from the input data.
- (3) In order to reduce the loss of image features during the process of pooling, we propose self-adaptive pooling (SAP).

## 2. Materials and Methods

### 2.1. Study Areas

In this study, two categories of Chinese high-spatial resolution remote-sensing images were used for urban water extraction: ZY-3 and GF-2 multispectral images. The detailed parameters of these images are provided in Table 1, considering the complex urban water network differences in China. The selected areas were located in Beijing, Tianjin and Chengdu. Four remote-sensing multispectral images acquired from the ZY-3 and Gaofeng-2 satellites having different scene sizes (2000 pixels  $\times$  1800 pixels to 2000 pixels  $\times$  1900 pixels) are analyzed in this study as shown in Figure 1. The study area covers three downtown districts which are surrounded by suburban water bodies such as lakes, ponds, narrow rivers and aquatic parks.

**Table 1.** Overview of Chinese ZY-3 and GF-2 multispectral datasets.

Satellite Parameters	ZY-3 Multispectral Imagery	GF-2 Multispectral Imagery
Product Level	1A	1A
Number of bands	4	4
Wavelength (nm)	Blue: 450–520; Green: 520–590	Red: 630–690; NIR: 770–890
Spatial resolution (m)	5.8	4
Radiometric resolution (bit)	1024	1024



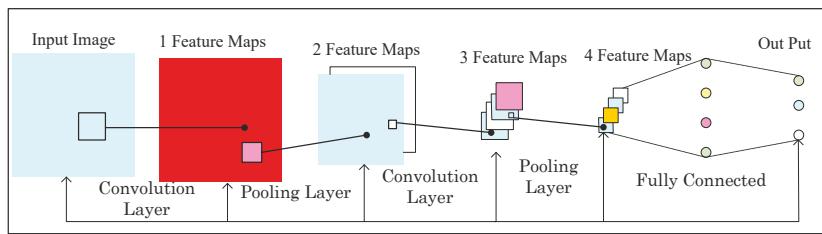
**Figure 1.** Study area and imagery materials. (a) ZY-3 multispectral imagery (Beijing, area coverage 2000 pixels × 1800 pixels), (b) ZY-3 multispectral imagery (Tianjin, area coverage 2000 pixels × 1800 pixels), (c) Gaofeng-2 multispectral imagery (Beijing, area coverage 2000 pixels × 1900 pixels), (d) ZY-3 multispectral imagery (Beijing, area coverage 2000 pixels × 1900 pixels).

All experiment images are Level 1A products, which have been adjusted for radiometric and geometric correction. Reference water mapping is manually digitized by a visual interpretation process of the high-resolution imagery with reference to Google Earth.

## 2.2. Self-Adaptive Pooling Convolutional Neural Networks (CNN) Architecture

A convolutional neural network (CNN) is a type of artificial neural network that draws inspiration from the biological visual cortex [21–23]. Compared with the shallow machine-learning algorithm, it has the advantages of strong applicability, parallel processing ability, and weight sharing, meaning global optimization training parameters are greatly reduced. CNN has become a hot topic in the field of deep learning [24]. The CNN architecture often consists of the input layer, convolution layer, pooling layer, full connection layer and output layer, as shown in Figure 2.

The convolutional layer consists of multiple feature maps which consist of multiple neurons. Each neuron is connected to the local area of the previous feature map by the convolution kernel [25]. The convolution kernel is a matrix of weights (such as  $3 \times 3$  or  $5 \times 5$  matrices for two-dimensional images). The convolutional layer extracts different features of the input layer through convolution operations. The first convolution layer extracts lower-level features such as edges, lines, corners, and higher-level convolution layers extract more advanced features.



**Figure 2.** The standard architecture of the convolutional neural network (CNN).

The input image is convolved in the convolutional and filtering layers. Generally, convolutional and filtering layers require an activation function to connect [26]. We use  $\mathbf{G}_i$  to represent the feature map of the  $i$ th layer of the convolutional neural network. The convolution process can be described as:

$$\mathbf{G}_i = f(\mathbf{G}_{i-1} \otimes \mathbf{W}_i + \mathbf{b}_i) \quad (1)$$

where,  $\mathbf{W}_i$  represents the weight feature vector of the  $i$ th convolution kernel, the operation symbol  $\otimes$  represents a convolution operation of the  $i$ th layer of the image and the  $i-1$ th layer of the image, and  $\mathbf{b}_i$  is the offset vector. Finally, the feature map  $\mathbf{G}_i$  of the  $i$ th layer is obtained by a linear activation function  $f(\bullet)$ .

There are two kinds of activation functions, one is a linear activation function, and the other is a non-linear activation function, such as sigmoid, hyperbolic and rectified functions. The rectified function is currently the most used in the literature because neurons, with a rectified function, work better to avoid saturation during the learning process, induce sparsity in the hidden units and do not face the gradient vanishing problem, which occurs when the gradient norm becomes smaller after successive updates in the back-propagation process [27]. So, in this paper, we use rectified linear unit (ReLU)  $f(x) = \max(x)$  as an activation function.

The pooling performs a sampling along the spatial dimensions of feature maps via a predefined function (e.g., maximum, average, etc.) on a local region. Although the high-level feature maps are more abstract, they lose a lot of detail due to the pooling operation. In order to reduce the loss of image features during the process of pooling, this paper presents an adaptive pooling model.

Due to the complexity of the objects in high-resolution images, the traditional pooling model cannot extract the image features very well. Therefore, this research takes two kinds of pooling areas in the pooling layer as shown in Figure 3. The blank space indicates that the pixel value is 0, the shaded area is composed of different pixel values, and A represents the maximum value area. The features of the whole feature map are mainly concentrated at A as shown in Figure 3a. If pooling is done with the average pooling model, the features of the entire feature map will be weakened. The features of the feature map are mainly distributed in A, B, C as shown in Figure 3b. In the case of the unknown relationship between A, B and C, the features of the entire feature map will be weakened by using the maximum pooling model. This will eventually affect the extraction accuracy of the water body in remote-sensing images.

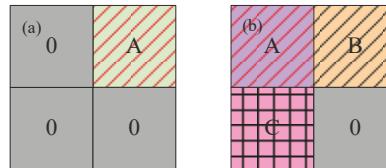
There are two main models of pooling layer: one is the max pooling model as shown in Equation (2), and the other is an average pooling model as shown in Equation (3). The feature map obtained by convolution layer is  $\mathbf{G}_{ij}$ , the size of the pooling area is  $c \times c$ , the pooling step length is  $c$ , and  $\mathbf{b}_i$  is the offset. The max pooling model can be expressed as:

$$\mathbf{F}_{ij} = \max_{i=1, j=1}^c (\mathbf{G}_{ij}) + \mathbf{b}_i \quad (2)$$

The average pooling model can be expressed as:

$$\mathbf{F}_{ij} = \frac{1}{c^2} \left( \sum_{i=1}^c \sum_{j=1}^c \mathbf{G}_{ij} \right) + \mathbf{b}_i \quad (3)$$

where,  $\max_{i=1,j=1}^c (\mathbf{G}_{ij})$  represents the max element from the feature map  $\mathbf{G}$  in the pooled region size  $c \times c$ .



**Figure 3.** Different pooling areas. (a) The features of the whole feature map are mainly concentrated at A; (b) The features of the feature map are mainly distributed in A, B, C.

In order to reduce the loss of image features during the process of pooling, this paper presents an adaptive pooling model according to the principle of interpolation, based on the maximum pool model and the average model. The model can adaptively adjust the pooling process through the pooling factors  $u$  in the complex pooled area. The expression is:

$$\mathbf{F}_{ij} = \frac{u}{c^2} \left( \sum_{i=1}^c \sum_{j=1}^c \mathbf{G}_{ij} \right) + (1-u) \max_{i=1,j=1}^c (\mathbf{G}_{ij}) + \mathbf{b}_i \quad (4)$$

where,  $u$  indicates pooling factor. The role of  $u$  is to dynamically optimize the traditional pooling model based on different pooled areas. The expression is:

$$u = \frac{a(b_{\max} - a)}{b_{\max}^2} \quad (5)$$

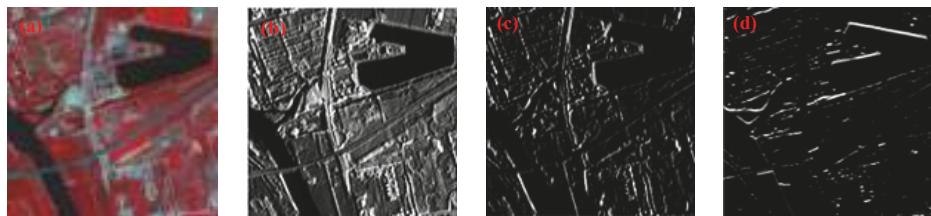
where,  $a$  is the average of all elements except for the max element in the pooled area,  $b_{\max}$  is the max element in the pooled area. The range of  $u$  is  $(0, 1)$ . The model takes into account the advantages of both the max pooling model and the average model. According to the characteristics of different pooling regions, the adaptive optimization model can be used to extract the features of the map as much as possible, so as to improve the removal accuracy of the convolution neural network.

In order to verify that the self-adaptive pooling model can reduce the loss of features in the pooling process that this paper proposes, an example image with the size of  $300 \times 300$  pixels is input into a simple network with a network structure of four layers. Figure 4a is the original image. Figure 4b is the feature map obtained from the self-adaptive pooling model, Figure 4c is the feature map obtained from the max pooling model, and Figure 4d is the feature map obtained from the average pooling model.

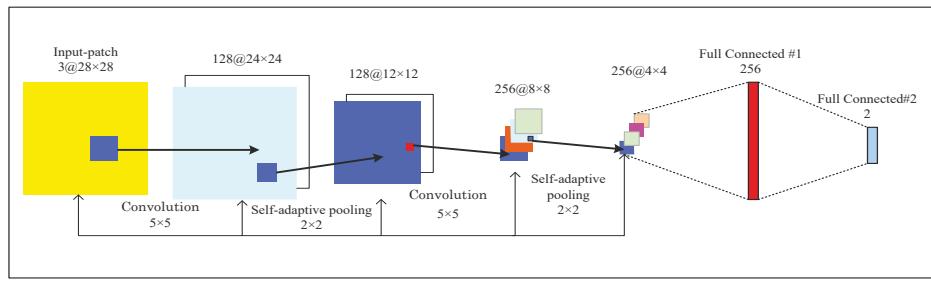
From Figure 4b-d, the feature map obtained from the adaptive pooling model has obvious features, but the max pooling model and the average pooling model weaken the image features.

As demonstrated in Figure 5, the overall architecture of the designed self-adaptive pooling convolutional neural network (SAPCNN) contains one input patch, two convolutional layers, two self-adaptive pooling layers, and two fully connected layers. An input patch is  $3@28 \times 28$ , consisting of three channels, each with a dimension of  $28 \times 28$ . The first convolution layer is  $128@24 \times 24$ , composed of 128 filters, followed by self-adaptive pooling of dimension  $2 \times 2$  resulting in  $128@12 \times 12$ . This process is followed by convolution layer and self-adaptive pooling; the convolution layer is  $256@8 \times 8$ , composed of 256 filters, and self-adaptive pooling is  $256@4 \times 4$ . All convolution layers have a stride of one pixel, and the size of filters is  $5 \times 5$ . In this paper, the output of the last fully

connected layer indicates the probabilities that the input patch belongs to water or no water. This means that the unit number of the output layer is two.



**Figure 4.** The feature map from the different pooling models. (a) the original image; (b) the feature map obtained from the self-adaptive pooling model; (c) the feature map obtained from the max pooling model; (d) the feature map obtained from the average pooling model.



**Figure 5.** Architecture of our designed CNN.

### 2.3. Pre-Processing

The convolutional neural networks extracts water bodies, but it does not guarantee continuous water bodies and water boundaries. Similarly, with building shadow, vegetation shadow, and mountain shadow, it does not guarantee compact contours and, hence, may misclassify water bodies. Therefore, a pre-processing step is required to reduce misclassified water bodies.

Superpixel refers to the adjacent image blocks with similar color and brightness characteristics [28]. It groups the pixels based on the similarities of features between pixels and obtains the redundant information of the image, which greatly reduces the complexity of subsequent image-processing tasks.

In this work, the image is segmented into superpixels, which are used as basic units to extract water bodies. As a widely used superpixel algorithm [29], the simple linear iterative clustering (SLIC) algorithm can output superpixels of good quality that are compact and roughly equally sized, but there are still some problems such as the fact that the number of superpixels should be designed artificially and the ultra-pixel edges are divided vaguely. However, because SLIC obtains initial cluster centers by dividing the image into several equal-size grids and its search space is limited to a local region [30], the superpixels produced cannot adhere to weak water boundaries well and the water bodies will be over-segmented. In this paper, the SLIC algorithm was improved by affinity propagation clustering and by expanding the search space.

#### 2.3.1. Color Space Transformation

Generally speaking, the color of water bodies is black and azure, with low reflectivity and high saturation. According to the features of the reflection spectrum of water bodies, a water body's region is prominent in B1, B2, and B4 to the data used in this study. Similar to the RGB color model, the color

space transformation to the hue, saturation, and intensity (HSI) color model is first performed using these 3 bands [31]. The transformation from the RGB to the HSI color model is expressed as follows:

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases} \quad (6)$$

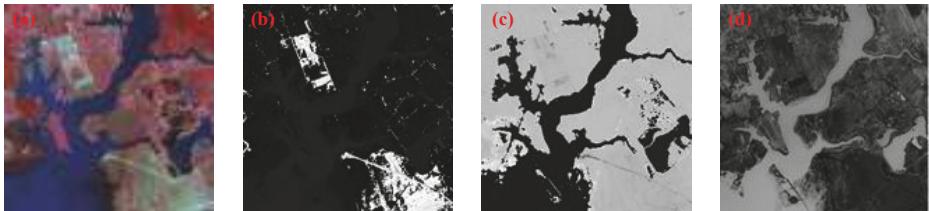
$$S = 1 - \frac{3 \times \min(R, G, B)}{R + G + B} \quad (7)$$

$$I = \frac{R + G + B}{3} \quad (8)$$

$$\theta = \cos^{-1} \left\{ \frac{[(R - G) + (R - B)]/2}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\} \quad (9)$$

where  $R$ ,  $G$ , and  $B$  are the values of B1, B2, and B4 channels of input remote sensing image.  $H$ ,  $S$ , and  $I$  are the values of hue, saturation, and intensity components in the HSI space.

Figure 6 shows the HSI color space of an example remote sensing image. Figure 6a is original RGB color image. Figure 6b is the intensity component image, Figure 6c is the hue component image, and Figure 6d is the saturation component image. From Figure 6b–d, it can be seen that the water bodies region is prominent in the  $H$  and  $S$  components. Therefore, the  $H$  and  $S$  components are used in our improved SLIC algorithm.



**Figure 6.** Hue, saturation, and intensity (HSI) color space of a remote-sensing image. (a) Original image. (b) Intensity component image. (c) Hue component image. (d) Saturation component image.

### 2.3.2. Adaptive Simple Linear Iterative Clustering (A-SLIC) Algorithm

The number of superpixels should be designed artificially as well as initial clustering. In this paper, the idea of affinity propagation algorithm is introduced to reduce the dependence of segmentation on initial conditions.

Usually, a weighted similarity measure combining color and spatial proximity is needed for the SLIC algorithm. In this study, the  $i$ th pixel and  $j$ th pixel space distance is expressed as follows:

$$d_{xy} = \sqrt{(x_i - x_{c_j})^2 + (y_i - y_{c_j})^2} \quad (10)$$

where,  $c_j$  is the  $j$ th pixel cluster center.

We define the color difference between  $i$ th and  $j$ th pixels as:

$$d_{xy} = \sqrt{(x_i - x_{c_j})^2 + (y_i - y_{c_j})^2} \quad (11)$$

We define the similarity measure between the  $i$ th pixel and  $j$ th cluster center  $c_j$  is expressed as follows:

$$d(i, j) = d_c + \frac{\alpha}{S} d_{xy} \quad (12)$$

where,  $S$  is the area of the  $j$ th cluster in the current loop. The  $\alpha$  parameter is used to control the relative importance between color similarity and spatial proximity.

By defining the attribution function (Equation (13)) and the attraction function (Equation (15)), the number and location of cluster centers are adjusted during the iteration to complete the superpixel adaptive segmentation. The attribution function reflects the possibility that pixel  $i$  attracts pixel  $j$  into its cluster [32]. The attribution function is expressed as:

$$\beta(i, j) = \begin{cases} \min_{i \neq j} \{0, \alpha(j, j) + \sum_{i' \neq i, j} \max[0, \alpha(i', j)]\} & i \neq j \\ \sum_{i' \neq j} \max[0, \alpha(i', j)] & i = j \end{cases} \quad (13)$$

The iteration relationship of the attribution function is expressed as:

$$\beta^t(i, j) = \begin{cases} \min_{i \neq j} \{0, \alpha^{t-1}(j, j) + \sum_{i' \neq i, j} \max[0, \alpha^{t-1}(i', j)]\} & i \neq j \\ \sum_{i' \neq j} \max[0, \alpha^{t-1}(i', j)] & i = j \end{cases} \quad (14)$$

The attraction function reflects the possibility of the  $j$  pixel attracting  $i$  pixel as its cluster [33]. The attraction function is expressed as:

$$\alpha(i, j) = s(i, j) - \max_{j' \neq j} \{\beta(i, j') + s(i, j')\} \quad (15)$$

The iteration relationship of the attraction function is expressed as:

$$\alpha^t(i, j) = s(i, j) - \max_{j' \neq j} \{\beta^{t-1}(i, j') + s(i, j')\} \quad (16)$$

where,  $s(i, j) = -d(i, j)$  is the similarity between  $i$  pixel and  $j$  pixel,  $s(i, j') = -d(i, j')$  is the similarity between  $i$  pixel and non- $j$  pixel, and  $t$  is the number of iterations.

Using both attraction and attribution functions, two types of messages are continuously transmitted to possible clustering centers to increase their likelihood of becoming cluster centers. So, the larger the sum of  $\alpha(i, j)$  and  $\beta(i, j)$ , the more likely the  $j$  pixel is a cluster center. In this case, the greater the probability that the  $i$  pixel belongs to this class, then the point is updated as a new cluster center. In order to reduce the computation complexity, the image is divided into  $n$  regions firstly and calculates  $\alpha(i, j)$  and  $\beta(i, j)$  in the local area. In this study, the main process of the A-SLIC algorithm is as follows:

**Step 1.** For an image containing  $M$  pixels, the size of the pre-divided region in this algorithm is  $N$ , then the number of regions is  $n$ . Each pre-divided area is labeled as  $\eta$ . In this paper,  $\alpha(i, j)$  and  $\beta(i, j)$  are defined zero, and  $t$  is defined one.

**Step 2.** HIS transformation is performed on each pre-divided area. In the  $\eta$ th region, according to Equation (10), the similarity between two pixels is calculated in turn.

**Step 3.** According to Equations (14) and (16), the sum of  $\beta^t(i, j)$  and  $\alpha^t(i, j)$  is calculated and the iteration begins.

**Step 4.** If  $\beta^t(i, j)$  and  $\alpha^t(i, j)$  no longer change or reach the maximum number of iterations, the iteration is terminated. The point where the sum of  $\beta^t(i, j)$  and  $\alpha^t(i, j)$  is max is regarded as the cluster center ( $R_i^\eta$ , where  $i = 1, 2, 3 \dots W_\eta$ ).

**Step 5.** Repeat steps 3 to 4 until the entire image is traversed, and adaptively determine the number of superpixels ( $R' = \sum_{\eta=1}^n W_\eta$ ). In this paper, the HSI value are the center of the pixel. Finally, complete the superpixel segmentation.

#### 2.4. Network Semi-Supervised Training and Extraction Waters

Convolution neural network training requires a large number of samples, but building a sample library requires a lot of time and manpower. In this paper, semi-supervised training is proposed. We use principal component analysis (PCA) to initialize the network structure [9], and then the entire network will be fine-tuned through the water label data.

Assume the input image has  $N$  scenes, its size is  $m \times n$ . The convolution filter size is  $g_1 \times g_2$ . In the  $i$ th scene of the training image, all the image blocks of size  $g_1 \times g_2$  are extracted and expressed as a vector form  $\mathbf{X}_i = \{x_{i,1}, x_{i,2}, x_{i,3} \dots, x_{i,nm}\}$ . Then the algorithm removes the mean of  $x_{i,nm}$  and expressed as a vector form  $\bar{\mathbf{X}}_i = \{\bar{x}_{i,1}, \bar{x}_{i,2}, \bar{x}_{i,3} \dots, \bar{x}_{i,nm}\}$ . So the image block of training data can be expressed as [9]:

$$\mathbf{X} = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3 \dots, \bar{\mathbf{x}}_n\} \in \mathbb{R}^{g_1 g_2 \times N_{nm}} \quad (17)$$

where,  $i$  is the number of the scene image.

The principal component analysis method can minimize the reconstruction error to solve the feature vector:

$$\begin{cases} \min_{V \in \mathbb{R}^{g_1 g_2 \times H_1}} \| \mathbf{X} - \mathbf{V} \mathbf{V}^T \mathbf{X} \|^2 \\ s.t. \mathbf{V}^T \mathbf{V} = \mathbf{I}_H \end{cases} \quad (18)$$

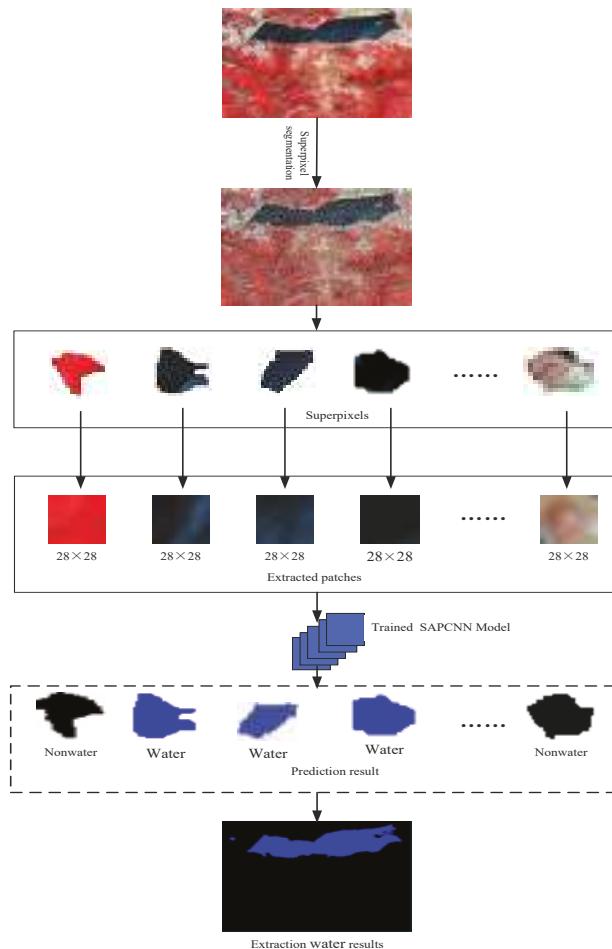
where,  $\mathbf{I}_H$  is a unit matrix,  $\mathbf{V}$  is the  $H$  feature vector of the covariance matrix  $(\mathbf{X} \mathbf{X}^T)$ .  $\mathbf{V}$  represents the main features of the input image block. The convolutional neural network  $\mathbf{W}_h$  filter were initialized by the principal component analysis, which can be expressed as follows:

$$\mathbf{W}_h = m_{g_1 g_2}(\mathbf{V}_h), h = 1, 2, 3, 4, 5 \dots H \quad (19)$$

where,  $m_{g_1 g_2}(\mathbf{V}_h)$  represents that vector  $\mathbf{V}$  is mapped to  $\mathbf{W}_h$ ,  $\mathbf{V}_h$  represents the  $h$ th main feature of the image.

In the training stage, we use a semi-supervised training method to train networks. First, the image of the training set is cut into the same size as the filter  $5 \times 5$  according to Equation (17) to create the training data set. According to Equations (18) and (19), the principal component analysis is used to obtain the initialized filter weight. Training is carried out by optimizing the logistic regression function using a stochastic gradient descent and mini-batch size of 128 with the momentum of 0.8. The training is regularized by weight decay set to 0.0001, and dropout regularization for all fully connected layers with the dropout ratio set to 0.1. The learning rate starts from 0.01 and is divided by 10 when the error plateaus. Finally, the algorithm fine-tunes the entire network through the water label data to complete the final network training. Through training, a SAPCNN classifier with two class predictions is generated for the extraction of urban water bodies.

In the extraction stage, superpixels are first obtained from the test remote-sensing image using the adaptive simple linear iterative clustering algorithm described in Section 2.3.2. Image patches with a size of  $28 \times 28$  centered at its geometric center pixel are extracted. Finally, image patches size of  $28 \times 28$  are inputted into the trained SAPCNN model. The procedure of water extraction is demonstrated in Figure 7.



**Figure 7.** Processing chain of water bodies extraction in the proposed framework.

## 2.5. Accuracy Assessment Method

Reference water mapping is manually digitized by a visual interpretation process of the high-resolution imagery with reference to Google Earth. We evaluate the algorithm performance for the water extraction in two aspects: (i) water classification accuracy, and (ii) water edge pixel extraction accuracy. Therefore, six metrics are used including overall accuracy (OA), producer's accuracy (PA), user's accuracy (UA), edge overall accuracy (EOA), edge omission error (EOE), and edge commission error (ECE).

Unit rates (Equation (20)) based on the confusion matrix are utilized to evaluate the final water maps produced using different method, including PA, UA and OA [4]. The definition is as follows:

$$PA = \frac{TP}{TP + FN}, UA = \frac{TP}{TP + FP}, OA = \frac{TP + TN}{T} \quad (20)$$

where,  $T$  is the total number of the pixels in the experimental remote sensing image, and  $TP, FN, FP$ , and  $TN$  are the pixels categorized by comparing the extracted water pixels with ground truth reference:

*TP*: true positives, i.e., the number of correct extraction pixels;

*FN*: false negatives, i.e., the number of the water pixels not extracted;

*FP*: false positives, i.e., the number of incorrect extraction pixels;

*TN*: true negatives, i.e., the number of no-water bodies pixels that were correctly rejected.

This paper defines the evaluation water edge pixel extraction accuracy as follows: (1) Firstly, obtain the boundary of water body by manual drawing. (2) The morphological expansion is performed in the water body boundary obtained in step (1) to create a buffer zone centered on the boundary line and having a radius of 3 pixels. (3) Finally, the pixels in the buffer area are judged. Suppose the total number of pixels in the buffer area is  $M$ , the number of correctly classified pixels is  $M_R$ , the number of missing pixels is  $M_O$ , and the number of false alarm pixels is  $M_C$ . Then *EOA*, *EOE* and *ECE* are defined as:

$$EOA = \frac{M_R}{M} \times 100\% \quad (21)$$

$$EOE = \frac{M_O}{M} \times 100\% \quad (22)$$

$$ECE = \frac{M_C}{M} \times 100\% \quad (23)$$

### 3. Experiments and Discussion

The proposed algorithm was implemented using python on a PC with a Intel(R) Xeon(R) E5-2630 CPU and GPU Nvidia Tesla M40 12G memory, and the designed SAPCNN algorithm was implemented through the software library tensorflow [10]. Our training dataset was collected from three ZY-3 multispectral images (south-west Beijing, China), two ZY-3 multispectral images (north-west Tianjin, China), and two Gaofeng-2 multispectral image (south-west Chengdu, China). In all experiments, the input patch was  $3@28 \times 28$ , and the output of the last fully connected layer indicated the probabilities of the input patch, belonging to water or non-water. This means that the unit number of the output layer is two. In this way, 8000 couples of patches are obtained from the training set, where the number of water and non-water patches are 3000 and 5000, respectively. For a test remote-sensing image, superpixels are first obtained by the adaptive simple linear iterative clustering algorithm. Then, image patches size of  $28 \times 28$  centered at its geometric center pixel are extracted from each superpixel and input into the trained SAPCNN model to predict the class of this superpixel. Finally, the extraction of the water bodies result from the test remote-sensing image is achieved by using the predictions of all its superpixels.

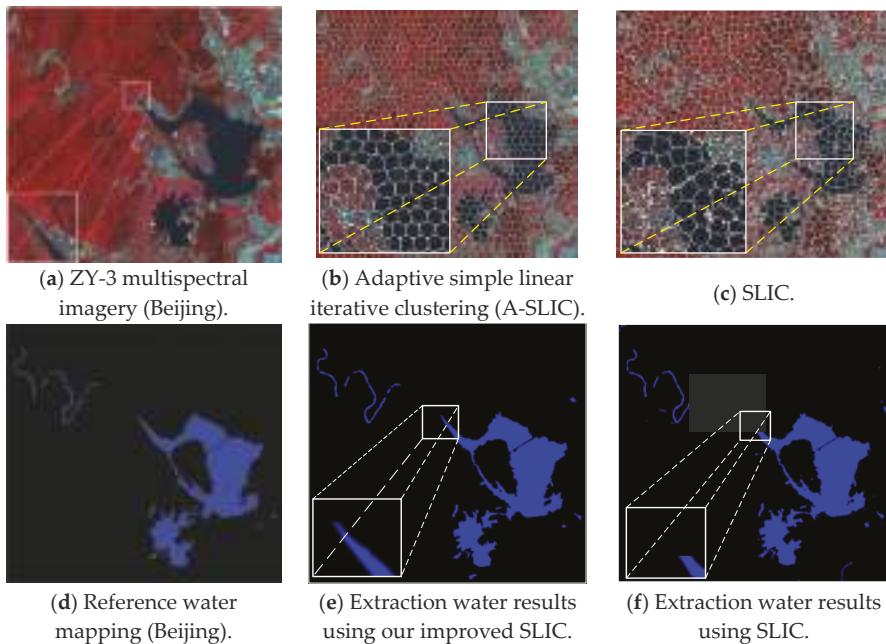
#### 3.1. Impact of the Superpixel Segmentation on the Performance of Water Mapping

In the proposed extraction water framework, the SLIC algorithm was used to cluster the remote sensing image into small regions, which is improved through affinity propagation clustering and expanding searching space. In order to verify the effectiveness of the A-SLIC method, we compared it with SLIC.

Figure 8b–c shows some superpixel segmentation results using different superpixel segmentation methods. Visual inspection of Figure 8b–c indicated that A-SLIC method and SLIC can obtain compact superpixels, but the A-SLIC method can obtain more regular superpixels than the SLIC method. The A-SLIC method can not only avoid over-segmentation in homogeneous sub-regions but can also obtain regular superpixels (in the white box).

We used water extraction results to evaluate the two superpixel methods. Figure 8e–f shows the extraction water results for the ZY-3 multispectral imagery (Beijing) in Figure 8a using the SAPCNN structure (see the Figure 4) combined with different superpixel methods, where Figure 8d represents the ground truth. It is obvious that all methods can extract most of the water bodies. However, for the blurry water boundaries and small water regions, our improved superpixel method can achieve more

accurate results through leading affinity propagation clustering, expanding searching space, and the produced superpixels are easier to adhere to blurry water boundaries.



**Figure 8.** Extraction water results using different superpixel segmentation methods. (a) ZY-3 multispectral imagery (Beijing); (b) Adaptive simple linear iterative clustering (A-SLIC); (c) SLIC; (d) Reference water mapping (Beijing); (e) Extraction water results using our improved SLIC; (f) Extraction water results using SLIC.

Six metrics (OA, PA, UA, EOA, EOE, and ECE) are used to evaluate the performance of water extraction using different superpixel segmentation methods. Table 2 shows the statistical results for SAPCNN framework using different superpixel segmentation methods.

From Table 2, compared with the SLIC method, the overall accuracy of water extraction and the PA are the highest for the superpixel segmentation method proposed in this paper for the pre-treatment of high-resolution images before water extraction. This is because the superpixel segmentation method introduced the idea of affinity propagation and adaptively determined the number of superpixels and the center of clustering, and the superpixel can well contain the water body boundary. Through the aforementioned visual evaluation and quantitative evaluation, it is verified that the SAPCNN method can effectively improve the water extraction accuracy and efficiency.

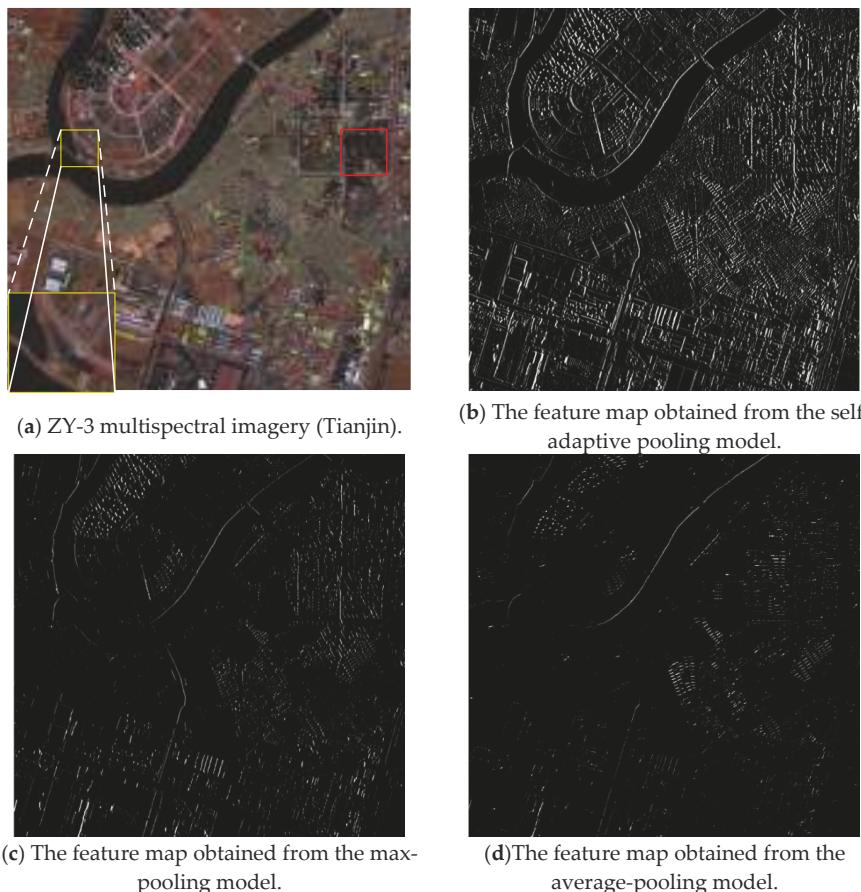
**Table 2.** Parameters of different superpixel segmentation water extraction method.

Image Name	Parameter	Our Method	SLIC
ZY-3 multispectral imagery(Beijing)	Overall accuracy (OA) (%)	99.29	97.29
	User's accuracy (UA) (%)	92.16	93.46
	Producer's accuracy (PA) (%)	87.19	82.06
	Edge overall accuracy (EOA) (%)	98.82	96.49
	Edge omission error (EOE) (%)	0.42	1.39
	Edge commission error (ECE) (%)	0.76	2.12

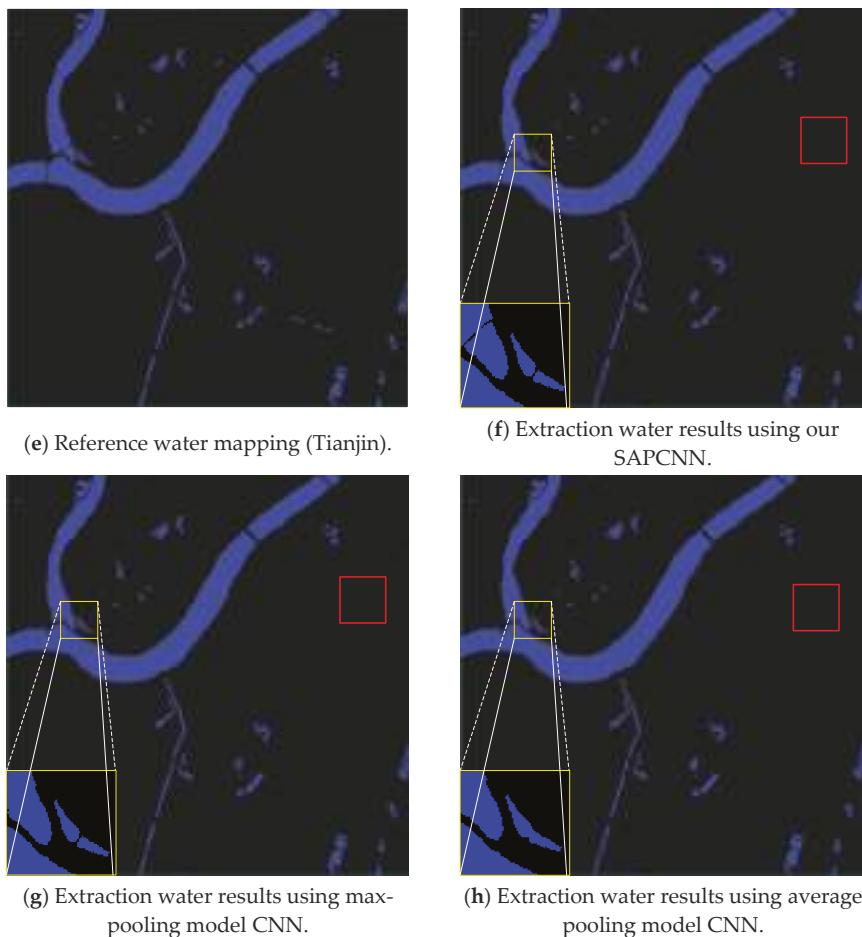
### 3.2. Comparison between Different Model CNN Architectures

In this paper, SAPCNN is designed to extract water bodies. We compare our SAPCNN with two different pooling model CNNs including a max pooling model CNN (the overall architecture of the designed max pooling CNN contains one input patch, two convolutional layers, two max pooling layers, and two fully connected layer) and an average pooling model CNN (the overall architecture of the designed max pooling CNN contains one input patch, two convolutional layers, two max-pooling layers, and two fully connected layer).

From Figure 9b–d, the feature map obtained from the adaptive pooling model has an obvious water boundary. While the max pooling model and the average pooling model weaken the water boundary. From Figure 9f–h, the CNNs is used to enhance the separation and difference between water and non-water, as well as to avoid spectral similarity between dark shadow and roads (in the red box). However, water extraction results using traditional pooling model CNNs blurred the water boundary.



**Figure 9. Cont.**



**Figure 9.** Extraction water results using different pooling models.

In order to evaluate objectively the edge-detection accuracy of the three kinds of water-extraction algorithms, three metrics are used: EOA, EOE, and ECE. Table 3 lists the boundary accuracy from different pooling model CNNs in which the self-adaptive pooling CNN method yields good results in two study areas. This is because the model can adaptively adjust the pooling process through the pooling factors in the complex pooled area. Therefore, the self-adaptive pooling CNN has more effective water extraction results compared with two tradition pooling CNNs.

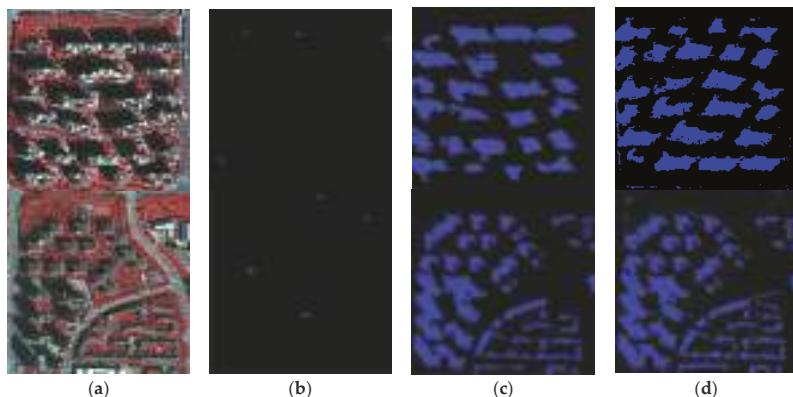
**Table 3.** Parameters of different pooling CNNs water extraction method.

Image Name	Parameter	Self-Adaptive Pooling + CNN	Max Pooling + CNN	Average Pooling + CNN
ZY-3 multispectral imagery(Tianjin)	EOA (%)	97.82	94.21	91.27
	EOE (%)	0.94	2.63	6.24
	ECE (%)	1.24	3.16	2.49

### 3.3. Distinguishing Shadow Ability of Different Methods

In order to verify that our improved methods can distinguish between water bodies and shadows, we compare our improved algorithm with SVM and NDWI. Two no-water bodies images were selected as experimental image. Figure 10 shows the water extraction results using SAPCNN, SVM and NDWI at the two no-water bodies' images.

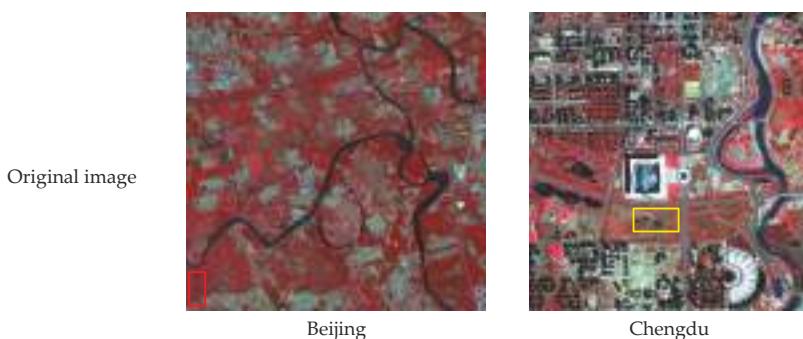
Visual inspection of Figure 10b indicated SAPCNN methods can distinguish between water bodies and shadows. But, visual inspection of Figure 10c,d indicated SVM and NDWI methods cannot distinguish between water bodies and shadows. Through the above mentioned visual evaluation, it is verified that SAPCNN method can effectively distinguish between water bodies and shadows.



**Figure 10.** Comparison of water extraction results using SAPCNN, SVM and NDWI. (a) No-water bodies images, (b) Extraction water results using SAPCNN, (c) Extraction water results using SVM, (d) Extraction water results using NDWI.

### 3.4. Comparison with Other Water Bodies Extraction Methods

In this paper, our algorithm uses three band images (B1, B2, and B4), the remote sensing is image first segmented into superpixels using the A-SLIC method, and the classes of these superpixels are then predicted using our designed SAPCNN model to obtain the water extraction result. We compare SAPCNN algorithm with SVM, the normalized difference water index (NDWI), and two extraction water methods in [34,35]. Figure 11 shows the water extraction results using SAPCNN, SVM, NDWI, and two extraction water methods in [34,35].



**Figure 11. Cont.**

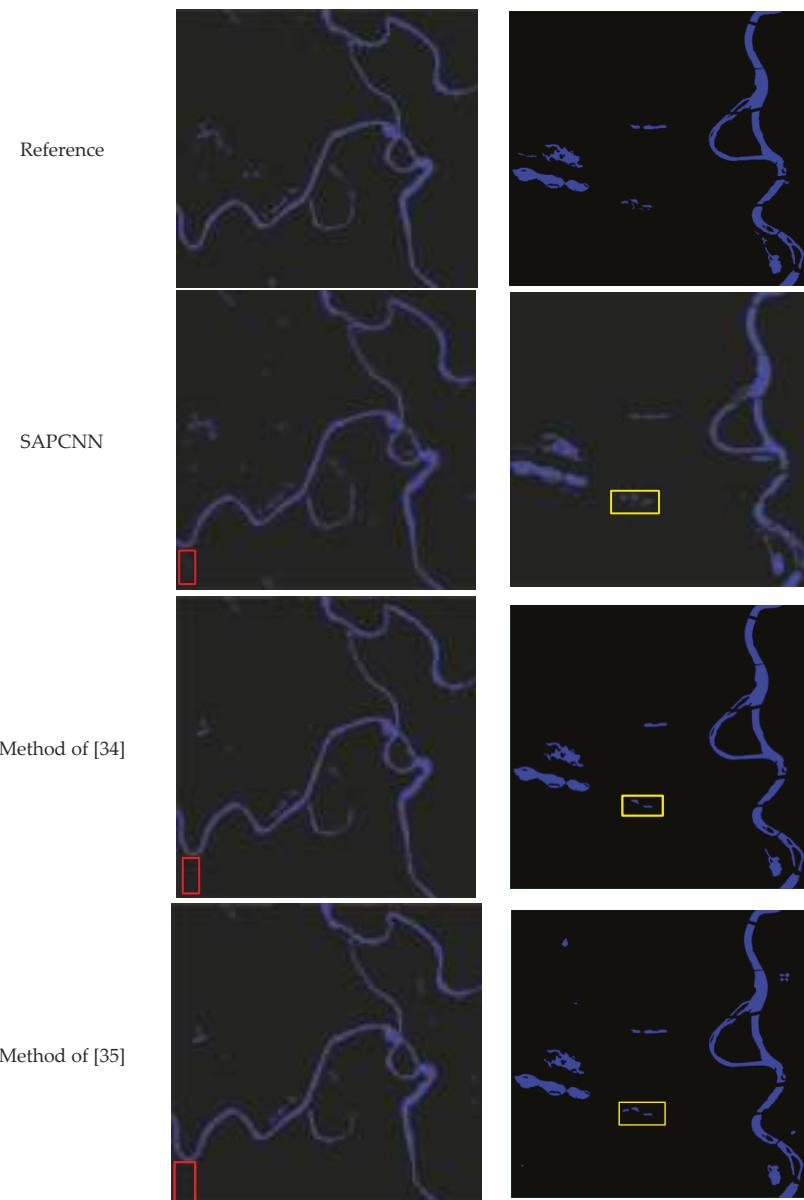
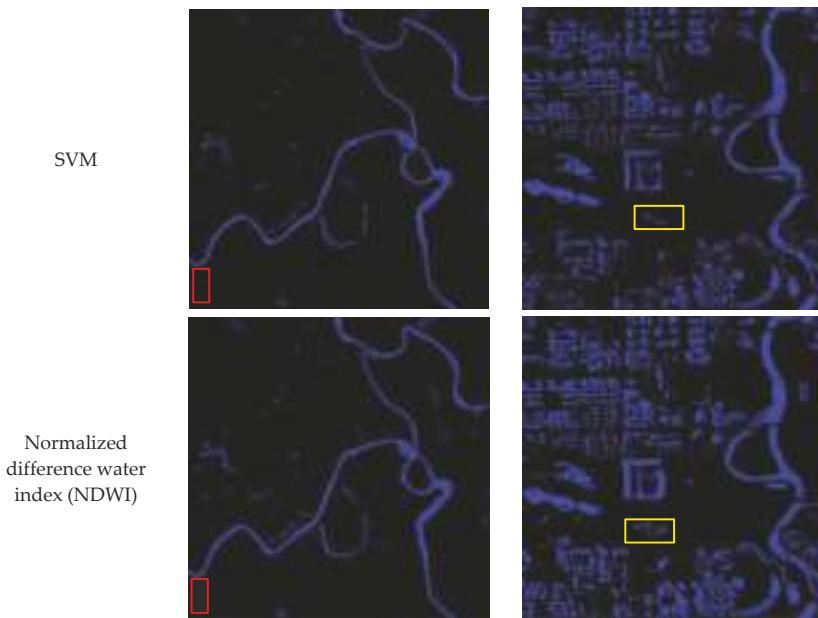


Figure 11. *Cont.*



**Figure 11.** Comparison of water extraction results using SAPCNN, SVM, NDWI and two extraction water methods in [34,35].

From Figure 11, it can be seen that SAPCNN successfully extracted most of the urban water bodies with complete shapes, while the extracted results by SVM, NDWI, the method of [34], and the method of [35] were incomplete. For example, SAPCNN extracted small ponds with complete shapes, whereas the results on the extracted ponds in NDWI, SVM and two water extraction methods in [34,35] were discontinuous (in the red box). From Figure 11, the water extraction results from the SAPCNN are visually cleaner, but the water extraction results from the NDWI and the SVM cannot distinguish between shadow pixels and water pixels. Here, the misidentified water pixels are found in residential areas, particularly in shadows and dark roads (in Chengdu). The extra shadow reduction procedure in the SAPCNN, the method of [34], and the method of [35] provides a larger improvement.

We evaluate the algorithm performance for the water extraction. Here, six metrics are used including OA, PA, UA, EOA, EOE and ECE. Table 4 shows statistical results for different water extraction algorithms on the test set. A good water extraction method has high values of OA and PA and low values of EOE and ECE.

Accuracy assessments (Table 4) indicate that the SAPCNN has a good accuracy when extracting urban water bodies. For the study area in Beijing, in the classification-level evaluation, we compared the SAPCNN with SVM, NDWI, the method of [34], and the method of [35], and found that the SAPCNN has a much higher overall accuracy (99.81%). In boundary evaluation of water bodies, in comparison to the SVM and the NDWI, our method has a much lower total edge omission and edge commission error (1.85%) and a much higher edge overall accuracy (98.15%). However, for the study area in Chengdu with a large number of shadow areas, we compared the NDWI with the SVM and the SAPCNN, and found that the NDWI has a much lower OA (69.17%) and PA (58.63%), and the SVM has a much higher total EOE and ECE (9.56%). The total ECE and EOE of SAPCNN was only 2.68% of that of SVM. The reason for this result is that the SVM and NDWI are more vulnerable to shadow pixels than the SAPCNN.

**Table 4.** Water mapping accuracy assessment results.

Study Area	Approach	OA	PA	UA	ECE	EOE	EOA
Beijing	SAPCNN	99.81%	90.24%	94.18%	1.13%	0.72%	98.15%
	Method of [34]	98.27%	89.21%	92.37%	2.61%	1.02%	96.37%
	Method of [35]	97.38%	91.32%	88.91%	2.10%	0.83%	97.07%
	SVM	88.21%	79.23%	81.54%	5.17%	1.34%	93.49%
	NDWI	89.36%	83.54%	80.09%	4.14%	1.27%	94.59%
Chengdu	SAPCNN	98.31%	92.33%	91.87%	1.64%	1.04%	97.32%
	Method of [34]	97.04%	91.79%	89.37%	3.03%	0.93%	96.04%
	Method of [35]	96.21%	90.37%	88.26%	2.95%	1.04%	96.01%
	SVM	71.23%	59.34%	63.54%	7.39%	2.17%	90.44%
	NDWI	69.17%	58.63%	65.27%	5.21%	3.57%	91.22%

#### 4. Conclusions

In this research, a novel water body extraction method based on deep learning is proposed for high-resolution remote-sensing images. The proposed method combines an enhanced superpixel method with deep learning to extract urban water bodies and distinguishes between shadow pixels and water pixels. The remote-sensing image is first segmented into superpixels using the A-SLIC method, and then a new CNN architecture is designed, which can mine high-level water features. The proposed method was tested for three different cities of China having different water-body types and topography, and results showed that the proposed method performed well with an accuracy of 98.31% to 99.81% and total EOE and ECE (2.68%). In addition, superpixel pre-processing reduces the size of feature maps of the SAPCNN and computation complexity. This study concludes that the proposed deep-learning methods can significantly improve urban surface water detection accuracy for high-resolution remote-sensing imagery.

**Author Contributions:** Yang Chen was responsible for the research design, experiment and analysis, and drafting of the manuscript. Rongshuang Fan collected the dataset. Xiucheng Yang and Jingxue Wang built the model. Aamir Latif wrote the paper. All authors reviewed the manuscript.

**Acknowledgments:** This work was supported by the Nation key R&D Program of China (No. 2016YFC0803100), the National Natural Science Foundation of China (No. 41101452), and the Doctoral Program Foundation of Institutions of Higher Education of China (No. 20112121120003).

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Fletcher, T.D.; Andrieu, H.; Hamel, P. Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art. *Adv. Water Res.* **2013**, *51*, 261–279. [[CrossRef](#)]
2. Rizzo, P. Water and Wastewater Pipe Nondestructive Evaluation and Health Monitoring: A Review. *Adv. Civ. Eng.* **2010**, *2010*, 818597. [[CrossRef](#)]
3. Byun, Y.; Han, Y.; Chae, T. Image fusion-based change detection for flood extent extraction using bi-temporal very high-resolution satellite images. *Remote Sens.* **2015**, *7*, 10347–10363. [[CrossRef](#)]
4. Yang, X.; Zhao, S.; Qin, X.; Zhao, N.; Liang, L. Mapping of Urban Surface Water Bodies from Sentinel-2 MSI Imagery at 10 m Resolution via NDWI-Based Image Sharpening. *Remote Sens.* **2017**, *9*, 596. [[CrossRef](#)]
5. Du, Y.; Zhang, Y.; Ling, F.; Wang, Q.; Li, W.; Li, X. Water bodies' mapping from Sentinel-2 imagery with Modified Normalized Difference Water Index at 10-m spatial resolution produced by sharpening the SWIR band. *Remote Sens.* **2016**, *8*, 354. [[CrossRef](#)]
6. Zhou, Y.; Luo, J.; Shen, Z.; Hu, X.; Yang, H. Multiscale water body extraction in urban environments from satellite images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4301–4312. [[CrossRef](#)]
7. Zeng, C.; Bird, S.; Luce, J.J.; Wang, J. A natural-rule-based-connection (NRBC) method for river network extraction from high-resolution imagery. *Remote Sens.* **2015**, *7*, 14055–14078. [[CrossRef](#)]

8. Zhang, Y.; Gao, J.; Wang, J. Detailed mapping of a salt farm from Landsat TM imagery using neural network and maximum likelihood classifiers: A comparison. *Int. J. Remote Sens.* **2007**, *28*, 2077–2089. [[CrossRef](#)]
9. Yan, Y.; Zhao, H.; Chen, C.; Zou, L.; Liu, X.; Chai, C.; Wang, C.; Shi, J.; Chen, S. Comparison of Multiple Bioactive Constituents in Different Parts of Eucommia ulmoides Based on UFLC-QTRAP-MS/MS Combined with PCA. *Molecules* **2018**, *23*, 643. [[CrossRef](#)]
10. Li, L.; Chen, Y.; Xu, T.; Liu, R.; Shi, K.; Huang, C. Super-Resolution Mapping of Wetland Inundation from Remote Sensing Imagery Based on Integration of Back-Propagation Neural Network and Genetic Algorithm. *Remote Sens. Environ.* **2015**, *164*, 142–154. [[CrossRef](#)]
11. McFeeters, S.K. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *Int. J. Remote Sens.* **1996**, *17*, 1425–1432. [[CrossRef](#)]
12. Huang, C.; Chen, Y.; Wu, J.; Li, L.; Liu, R. An evaluation of Suomi NPP-VIIRS data for surface water detection. *Remote Sens. Lett.* **2015**, *6*, 155–164. [[CrossRef](#)]
13. Xu, H. Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *Int. J. Remote Sens.* **2006**, *27*, 3025–3033. [[CrossRef](#)]
14. Feyisa, G.L.; Meilby, H.; Fensholt, R.; Proud, S.R. Automated Water Extraction Index: A new technique for surface water mapping using Landsat imagery. *Remote Sens. Environ.* **2013**, *140*, 23–35. [[CrossRef](#)]
15. Katz, D. Undermining demand management with supply management: Moral hazard in Israeli water policies. *Water* **2016**, *8*, 159. [[CrossRef](#)]
16. Kang, L.; Zhang, S.; Ding, Y.; He, X. Extraction and preference ordering of multireservoir water supply rules in dry years. *Water* **2016**, *8*, 28. [[CrossRef](#)]
17. Niroumand-Jadidi, M.; Vitti, A. Reconstruction of river boundaries at sub-pixel resolution: Estimation and spatial allocation of water fractions. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 383. [[CrossRef](#)]
18. Vieira, S.; Pinaya, W.H.L.; Mechelli, A. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. *Neurosci. Biobehav. Rev.* **2017**, *74*, 58–75. [[CrossRef](#)] [[PubMed](#)]
19. Singh, P.; Verma, A.; Chaudhari, N.S. Deep Convolutional Neural Network Classifier for Handwritten Devanagari Character Recognition. In *Information Systems Design and Intelligent Applications*; Springer: New Delhi, India, 2016.
20. Zhou, F.-Y.; Jin, L.-P.; Dong, J. Review of Convolutional Neural Network. *Chin. J. Comput.* **2017**, *40*, 1229–1251.
21. Hu, F.; Xia, G.-S.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [[CrossRef](#)]
22. Chen, J.; Wang, C.; Ma, Z.; Chen, J.; He, D.; Ackland, S. Remote Sensing Scene Classification Based on Convolutional Neural Networks Pre-Trained Using Attention-Guided Sparse Filters. *Remote Sens.* **2018**, *10*, 290. [[CrossRef](#)]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012.
24. Vedeld, A.; Lenc, K. MatConvNet: Convolutional neural networks for MATLAB. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015.
25. Yang, L.; Tian, S.; Yu, L.; Ye, F.; Qian, J.; Qian, Y. Deep learning for extracting water body from Landsat imagery. *Int. J. Innov. Comput. Inf. Control* **2015**, *11*, 1913–1929.
26. Yang, J.; Yang, G. Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer. *Algorithms* **2018**, *11*, 28. [[CrossRef](#)]
27. Pouliot, D.; Latifovic, R.; Pasher, J.; Duffe, J. Landsat Super-Resolution Enhancement Using Convolution Neural Networks and Sentinel-2 for Training. *Remote Sens.* **2018**, *10*, 394. [[CrossRef](#)]
28. Csillik, O. Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels. *Remote Sens.* **2017**, *9*, 243. [[CrossRef](#)]
29. Li, Z.; Chen, J. Superpixel segmentation using linear spectral clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1356–1363.
30. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [[CrossRef](#)] [[PubMed](#)]

31. Zollhöfer, M.; Izadi, S.; Rehmann, C.; Zach, C.; Fisher, M.; Wu, C.; Fitzgibbon, A.; Loop, C.; Theobalt, C.; Stamminger, M. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graph.* **2014**, *33*, 156. [[CrossRef](#)]
32. Li, H.; Liu, J.; Liu, R.W.; Xiong, N.; Wu, K.; Kim, T.-H. A Dimensionality Reduction-Based Multi-Step Clustering Method for Robust Vessel Trajectory Analysis. *Sensors* **2017**, *17*, 1792. [[CrossRef](#)] [[PubMed](#)]
33. Guangyun, Z.; Xiuping, J.; Jiankun, H. Superpixel-based graphical model for remote sensing image mapping. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5861–5871.
34. Isikdogan, F.; Bovik, A.C.; Passalacqua, P. Surface water mapping by deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4909–4918. [[CrossRef](#)]
35. Yu, L.; Wang, Z.; Tian, S.; Ye, F.; Ding, J.; Kong, J. Convolutional neural networks for water body extraction from landsat imagery. *Int. J. Comput. Intell. Appl.* **2017**, *16*, 1750001. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Data Pre-Analysis and Ensemble of Various Artificial Neural Networks for Monthly Streamflow Forecasting

Jianzhong Zhou <sup>1,2,\*</sup>, Tian Peng <sup>1,2,\*</sup>, Chu Zhang <sup>1,2</sup> and Na Sun <sup>1,2</sup>

<sup>1</sup> School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; zhangchuhust@foxmail.com (C.Z.); husthydro08@126.com (N.S.)

<sup>2</sup> Hubei Key Laboratory of Digital Valley Science and Technology, Wuhan 430074, China

\* Correspondence: jz.zhou@mail.hust.edu.cn (J.Z.); husthydtop@126.com (T.P.); Tel.: +86-(0)27-8754-3127 (J.Z.)

Received: 28 March 2018; Accepted: 7 May 2018; Published: 13 May 2018

**Abstract:** This paper introduces three artificial neural network (ANN) architectures for monthly streamflow forecasting: a radial basis function network, an extreme learning machine, and the Elman network. Three ensemble techniques, a simple average ensemble, a weighted average ensemble, and an ANN-based ensemble, were used to combine the outputs of the individual ANN models. The objective was to highlight the performance of the general regression neural network-based ensemble technique (GNE) through an improvement of monthly streamflow forecasting accuracy. Before the construction of an ANN model, data preanalysis techniques, such as empirical wavelet transform (EWT), were exploited to eliminate the oscillations of the streamflow series. Additionally, a theory of chaos phase space reconstruction was used to select the most relevant and important input variables for forecasting. The proposed GNE ensemble model has been applied for the mean monthly streamflow observation data from the Wudongde hydrological station in the Jinsha River Basin, China. Comparisons and analysis of this study have demonstrated that the denoised streamflow time series was less disordered and unsystematic than was suggested by the original time series according to chaos theory. Thus, EWT can be adopted as an effective data preanalysis technique for the prediction of monthly streamflow. Concurrently, the GNE performed better when compared with other ensemble techniques.

**Keywords:** monthly streamflow forecasting; artificial neural network; ensemble technique; phase space reconstruction; empirical wavelet transform

## 1. Introduction

Streamflow forecasting has been one of the key issues in hydrology in recent decades. Enhancing streamflow forecasting accuracy is of great significance to various aspects of hydrological system such as water allocation, flood control, and disaster relief. In recent decades, numerous methods and hydrological models have been studied to obtain accurate streamflow predictions. These methods can be grouped into two categories: conceptual models and empirical models [1]. Conceptual models, also known as physically based models, are designed to simulate the physical mechanism of hydrological processes [2]. However, because of insufficient data collection both in space and time for conceptual models, these models may not be feasible for streamflow forecasting [3]. On the other hand, empirical models are data-driven models which are built using historical information contained in the hydrological time series as opposed to the physical processes of a certain catchment [4–6]. The various empirical models involved in hydrological forecasting predominantly include time series models, machine learning methods, and hybrid methods. Time series models, especially auto-regressive moving average models, have been one of the most popular methodologies for streamflow forecasting over the last decades. However, the results of the previous studies have shown that time series models only

provide satisfactory results when the series are either linear or near-linear; they do not perform well with non-linear series [1,7].

As a result of this limitation of time series models, in recent years, various machine learning methods have been applied in the forecasting of non-linear hydrological systems. Among the various machine learning methods, artificial neural networks (ANNs), which include backpropagation neural network (BPNN), radial basis function (RBF) neural network, generalized regression neural network (GRNN), Elman neural network, and multi-layer feed-forward (MLFF) network, are among the most popular techniques for hydrological time series forecasting. Chen et al. [8] applied three different ANN models, namely a MLFF network, a RBF network, and a GRNN, to predict the streamflow, and copula-entropy (CE) was utilized to identify the inputs of the networks. Results showed that the MLFF network with the CE method obtained better results in comparison with traditional linear correlation analysis. Chang et al. [9] successfully introduced BPNN, Elman neural network, and NARX network into the forecasting of one-to six-steps ahead of floodwater storage pond water levels. The results indicated that the proposed NARX model could be beneficial for the control of urban floods. Hosseini-Moghari and Araghinejad [10] utilized the recursive and direct versions of Multi-Layer Perceptron (MLP), RBF and GRNN neural networks for forecasting droughts at short-, mid-, and long-term time scales, respectively. Results showed that the recursive models obtained better results at smaller time scales of the Standard Precipitation Index while the direct models showed better performance at longer time scales.

Numerous successes have been obtained in the applications of ANNs for time series forecasting; rooms exists to improve single ANN method performance. One trend to enhance the performance of ANN models for time series forecasting is to employ data-preprocessing techniques [11,12]. Wang et al. [13] presented a hybrid approach which combined ensemble empirical mode decomposition (EEMD) and artificial neural networks for medium and long-term runoff forecasting. Results of this study indicated that EEMD could enhance forecasting accuracy of medium and long-term runoff time series. Zhu et al. [14] developed signal decomposition techniques, including discrete wavelet transform (DWT) and empirical mode decomposition (EMD), to improve the forecasting accuracy of the support vector machine (SVR) models for monthly streamflow prediction. Results have shown both EMD and DWT can improve the forecasting accuracy of monthly streamflow, while DWT performed better EMD in enhancing the forecasting accuracy of the SVM model. Seo et al. [15] compared and evaluated three hybrid models for forecasting daily river stages: the wavelet package-ANN (WPANN) model, the wavelet package-adaptive neuro-fuzzy inference system (WPANFIS) model, and the wavelet package-SVM (WPSVM) model. The results obtained indicated that the WPANFIS models provided better prediction results than the WPANN and WPSVM models. Although WT- and EMD-based data preprocessing techniques have shown their efficiency in promoting the performance of machine learning forecasting methods, the performance of WT is sensitive to the selection of mother wavelets and EMD is likely to be encountered with the mode mixing problem. The Empirical Wavelet Transform (EWT) proposed by Gilles [16] solves these problems. The efficiency of EWT as a data preanalysis method to improve the forecasting accuracy of machine learning methods has been demonstrated in Hu and Wang [17] and Wang and Hu [18] for mean half-hour wind speed and mean 15 min wind speed forecasting, respectively.

To construct an ANN model for streamflow forecasting, one of the most important steps is to determinate appropriate input vectors. Numerous studies have confirmed and verified the existence of chaotic behavior in hydrological time series as that generated by the underlying stochastic processes, the phase space reconstruction (PSR) method has been utilized as an alternative approach to select relevant and important input variables for ANN models [19]. Guo et al. [20] introduced the PSR method to determine the inputs of the SVR streamflow prediction model to overcome the drawbacks in the empirical judgment within the structure of the forecasting model. Hu et al. [21] investigated the cross-scale chaotic behaviors of the runoff processes in an inland river of central Asia by using the

PSR technique and chaos theory. Ouyang et al. [22] successfully introduced the PSR method in the construction of an input matrix for SVR models for forecasting monthly rainfall.

Several papers have studied the employment of a single ANN [23], while other have compared the performance of different ANN architectures. Because an ensemble model often can obtain more accurate results than its constituent components, employing ensemble techniques has become a popular topic in recent years to enhance the generalizability and reliability of ANNs. Ensemble techniques have already been successfully applied to numerous time series predictions such as wind and solar power forecasting [24] as well as heating energy consumption predictions [25]. In this study, the objective was to investigate the efficiency of various ensemble techniques including simple averaging ensemble (SAE), weighted averaging ensemble (WAE), and GRNN-based ensemble (GNE) through the combination of the outputs of various single ANN models. Additionally, to take advantage of both the superior performance of EWT in data preanalysis and the PSR technique in selecting input vectors, the EWT and PSR methods were exploited in the proposed ensemble forecasting models. Mean monthly streamflow observation data from Wudongde hydrological station in Jinsha River Basin, China was used to demonstrate the efficiency of the proposed GNE ensemble model.

## 2. Methodology

### 2.1. Artificial Neural Networks

#### 2.1.1. Radial Basis Function Neural Network

Radial basis function neural network is type of multilayer and feed-forward neural network (FNN) [26]. Similar to traditional ANNs, the RBF neural network consists of three layers which include: an input layer which composed of input variables, a hidden layer where the input variables are transformed by a nonlinear function, and a linear output layer which produces the network response. In comparison with the most commonly used sigmoidal functions employed by a FNN, the hidden layer of a RBF neural network uses Gaussian transfer functions as activation functions. The Gaussian activation function can be written as:

$$\phi_i(x) = \exp\left(-\|x - c_i\|^2 / 2q_i^2\right) \quad (1)$$

where  $x = [x_1, x_2, \dots, x_n]^T$  is the input vector with  $N$  dimensions,  $c_i = [c_{i1}, c_{i2}, \dots, c_{in}]^T$  is the center of the  $i$ th neuron in the hidden layer,  $q_i$  is the width of the Gaussian function, and  $\|\cdot\|$  is the Euclidean Norm. The response of the  $j$ th node in the output layer can be written as

$$y_j(x) = \sum_{i=1}^h w_{ij}\phi_i(x) \quad j = 1, 2, \dots, m \quad (2)$$

where  $w_{ij}$  is the connecting weight between the  $i$ th hidden node and the  $j$ th output node.

#### 2.1.2. Extreme Learning Machine

Extreme Learning Machine (ELM) developed by Huang et al. [27] is a new type of single hidden layer feed-forward network (SLFN). Given a set of  $N$  samples  $(x_i, t_i)$ ,  $i = 1, 2, \dots, N$ , the ELM network with  $L$  hidden neurons and activation function  $g$  can be referred to as:

$$f_L = \sum_{i=1}^L \beta_i g(\mathbf{a}_i \cdot \mathbf{x}_j + b_i) \quad (3)$$

where  $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbf{R}^n$ ;  $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbf{R}^m$ ,  $\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{ni}]$  is the weight vector that connects the  $n$  input neurons with the  $i$ th hidden neuron;  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  is the weight vector that connects the  $i$ th hidden neuron with the  $m$  output neurons;  $b_i$  is the bias.

Equation (3) can be abbreviated as:

$$H\beta = T \quad (4)$$

where

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(a_1 \cdot x_1 + b_1) & \cdots & g(a_L \cdot x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(a_1 \cdot x_N + b_1) & \cdots & g(a_L \cdot x_N + b_L) \end{bmatrix}_{N \times L} \quad (5)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (6)$$

and  $H$  is the output matrix of the hidden layer. The output weights of ELM can be obtained by calculating the least square solution of the following equation:

$$\|H\hat{\beta} - T\| = \|HH^\dagger T - T\| = \min_{\beta} \|H\beta - T\| \quad (7)$$

The least square solution can be given as:

$$\hat{\beta} = H^\dagger T \quad (8)$$

where  $H^\dagger$  denotes the Moore-Penrose generalized inverse of  $H$  [27].

### 2.1.3. Elman Neural Network

Elman neural network is a kind of dynamic recurrent neural network [28]. In addition to the input layer, hidden layer, and output layer, an Elman network has a special recurrent layer which connects every input unit to a hidden unit; every hidden unit has a corresponding time delay [9]. The recurrent layer is used to store the output information of the hidden layer within a certain time delay; that information is then used as the input for the hidden layer. Therefore, the outputs of the Elman network depend not only on the preset inputs but also on the previous states of the hidden units [9]. In this study, the Elman network was trained using a gradient descent with momentum; the transfer functions of hidden and output layers were of sigmoid and linear types, respectively.

### 2.1.4. General Regression Neural Network

The GRNN was first introduced by Specht [29] and was a variation of RBF. Assuming a random vector  $X$  and the joint continuous probability density function  $p(X, y)$  is known, the regression of  $Y$  on  $x$  can be given by [29,30]:

$$E[y|X] = \frac{\int_{-\infty}^{\infty} yp(X, y) dy}{\int_{-\infty}^{\infty} p(X, y) dy} \quad (9)$$

where  $E[y|X]$  is the conditional expect of the output  $y$  given the input vector  $X$ . The joint density  $p(X, y)$  is usually unknown and can be estimated by the Gaussian kernel estimator:

$$\hat{p}(X, y) = \frac{1}{n(2\pi)^{(d+1)/2}\sigma^{(d+1)}} \sum_{j=1}^n \exp\left[-\frac{D_j^2}{2\sigma^2}\right] \exp\left[-\frac{(y - y_j)^2}{2\sigma^2}\right] \quad (10)$$

where  $D_j^2 = (X - X^j)^T (X - X^j)$ ,  $n$  is the number of observations,  $d$  is the dimension of  $X$ , and  $\sigma$  is the smoothing parameter.

According to Equations (9) and (10), the general version of GRNN can be obtained as follows:

$$\hat{y}(X) = \frac{\sum_{j=1}^n y^j \exp\left(-\frac{D_j^2}{2\sigma^2}\right)}{\sum_{j=1}^n \exp\left(-\frac{D_j^2}{2\sigma^2}\right)} \quad (11)$$

where  $\hat{y}(X)$  is the probability estimate function of  $y(x)$ .

## 2.2. Phase Space Reconstruction

To transfer a one-dimension time series into a multi-dimensional phase-space, Packard et al. [31] proposed a PSR method. The PSR method can fully uncover the hidden information of the time series. For a given time series  $x(i), i = 1, 2, \dots, n$ , the key of the PSR method is to find the embedded dimension  $m$  and the parameter of time delay  $\tau$ , such that

$$\begin{cases} X_k = \{x(k), x(k + \tau), \dots, x(k + (m - 1)\tau)\} \\ Y_k = \{x(k + m\tau)\} \end{cases} \quad k = 1, 2, \dots, n - (m - 1)\tau \quad (12)$$

where  $X_k$  represents the  $k$ th phase point of the input vector and  $Y_k$  represents the  $k$ th phase point of the output vector.

In this study, the auto-correction function (ACF) value and Average Mutual Information (AMI) were utilized to determine the time delay  $\tau$ ; the Cao method [32] was utilized to determine the embedded dimension  $m$ . Generally, the time delay  $\tau$  is selected when the ACF first passes through zero value or the AMI arrives at the first minimum [33]. The value of  $m$  is determined according to Cao [32], once  $E1$  stops changing when  $d$  is greater than some value  $d_0$ . Subsequently, the minimum embedding dimension is selected as  $d_0 + 1$ . Readers may refer to Cao [32] for more information about the Cao method.

Only time series with chaotic characteristics can obtain accurate forecasting results using chaotic theories. After the time delay and embedded dimension are determined, the chaotic characteristics of the time series should be confirmed. The most commonly used method is the maximum Lyapunov exponent method. If the maximum Lyapunov exponent  $\lambda$  of the time series is positive, then the time series shows chaos features. Otherwise,  $\tau$  and  $m$  needs to be redefined. In this study, the small data sets method proposed by Rosenstein et al. [34] was used to calculate the maximum Lyapunov exponent of a time series.

## 2.3. Empirical Wavelet Transform

Empirical wavelet transform developed by Gilles [16] is used to decompose the given signal into a collection of amplitude modulated–frequency modulated (AM-FM) signals according to the information contained in the Fourier spectrum of the signal. For a given time series  $x(t)$ , the decomposition processes using EWT can be described in the following five steps.

Firstly, the Fourier spectrum  $F(\omega)$  of the original time series was calculated using Fast Fourier Transform Algorithm;

Secondly, the boundaries  $\omega_i$  were determined by proper segmentation of the Fourier spectrum:

$$\omega_i = \frac{f_i + f_{i+1}}{2} \text{ for } 1 \leq i \leq N - 1 \quad (13)$$

where  $\{f_i\}, i = 1, 2, \dots, N$  denotes the frequencies corresponding to the local maxima and  $f_0 = 0$ ,  $f_N = \pi$ .

Thirdly, the empirical wavelets  $\psi_i(\omega)$  and scaling function  $\varphi_1(\omega)$  were constructed:

$$\psi_i(\omega) = \begin{cases} 1 & , if (1+\gamma)\omega_i \leq |\omega| \leq (1-\gamma)\omega_{i+1} \\ \cos\left[\frac{\pi}{2}\beta\left(\frac{1}{2\gamma\omega_{i+1}}(|\omega| - (1-\gamma)\omega_{i+1})\right)\right] & , if (1-\gamma)\omega_{i+1} \leq |\omega| \leq (1+\gamma)\omega_{i+1} \\ \sin\left[\frac{\pi}{2}\beta\left(\frac{1}{2\gamma\omega_i}(|\omega| - (1-\gamma)\omega_i)\right)\right] & , if (1-\gamma)\omega_i \leq |\omega| \leq (1+\gamma)\omega_i \\ 0 & , otherwise \end{cases} \quad (14)$$

$$\varphi_1(\omega) = \begin{cases} 1 & , if |\omega| \leq (1-\gamma)\omega_1 \\ \cos\left[\frac{\pi}{2}\beta\left(\frac{1}{2\gamma\omega_1}(|\omega| - (1-\gamma)\omega_1)\right)\right] & , if (1-\gamma)\omega_1 \leq |\omega| \leq (1+\gamma)\omega_1 \\ 0 & , otherwise \end{cases} \quad (15)$$

where  $\beta(x) = x^4(35 - 84x + 70x^2 - 20x^3)$ ,  $\gamma < \min_i\left(\frac{\omega_{i+1}-\omega_i}{\omega_{i+1}+\omega_i}\right)$ .

Fourthly, and the approximate and detail coefficients were calculated:

$$W_x(i, t) = \langle x(t), \psi_i(t) \rangle = \int x(\tau)\psi_i(\tau-t)d\tau = F^{-1}[x(\omega)\psi_i(\omega)] \quad (16)$$

$$W_x(1, t) = \langle x(t), \varphi_1(t) \rangle = \int x(\tau)\varphi_1(\tau-t)d\tau = F^{-1}[x(\omega)\varphi_1(\omega)] \quad (17)$$

Finally, the original signal was reconstructed to obtain different modes:

$$x(t) = W_x(1, t) \times \varphi_1(t) + \sum_{i=2}^N W_x(i, t) \times \psi_i(t) \quad (18)$$

Readers may refer to Gilles [16] for more information about EWT.

#### 2.4. Ensemble Techniques

The generalizability and reliability of an ANN model can be often improved by appropriate ensemble techniques. Because the forecasting results of the individual models can vary in different data points, the error of the individual networks can be compensated by combining the outputs [25]. In this study, three kinds of ensemble techniques, namely, SAE, WAE, and GNE, were introduced to combine the outputs of different ANNs and were studied.

##### 2.4.1. Simple Averaging Ensemble

The SAE takes advantage of the concept of the arithmetic mean. Consider an ANN ensemble model with K sub-ANNs, the output of the SAE model can be defined as:

$$\bar{y} = \frac{1}{K} \sum_{k=1}^K y_i^k i = 1, 2, \dots, N. \quad (19)$$

where  $y_i^k$  is the output of the  $k$ th sub-ANN and  $N$  denotes the length of the data set.

##### 2.4.2. Weighted Averaging Ensemble

In the WAE model, the weighted means of the outputs of the sub-ANNs constructed the ensemble output. The output of the WAE model with K sub-ANNs can be defined as:

$$\bar{y}_i = \sum_{k=1}^K w_{ki} y_i^k i = 1, 2, \dots, N. \quad (20)$$

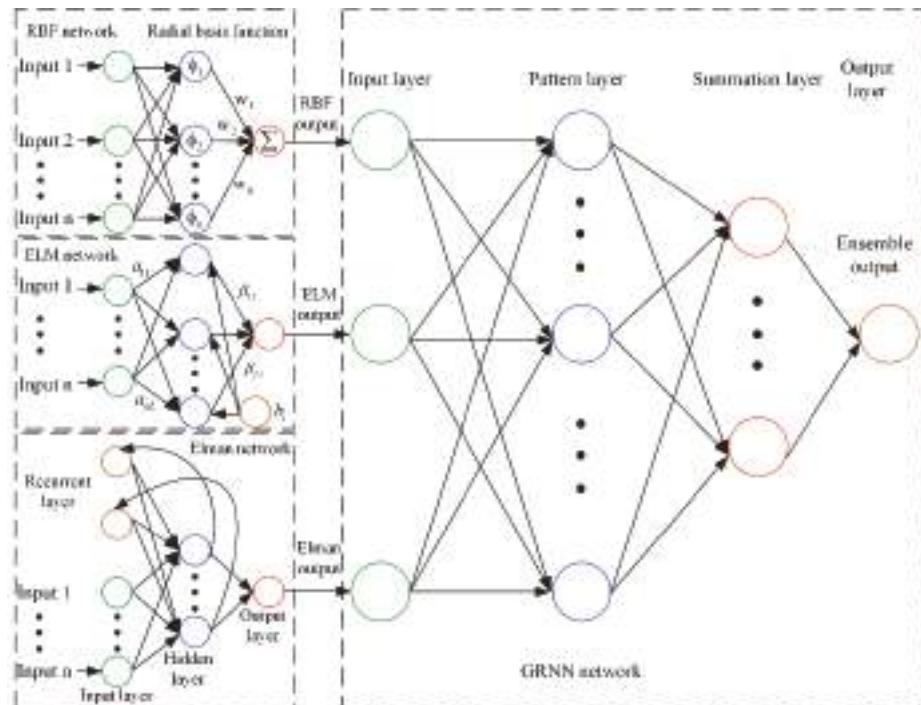
where  $w_{ki}$  denotes the weight of the  $k$ th model at point  $i$  and  $\sum_{k=1}^K w_{ki} = 1, w_{ki} \geq 0$ . The specific value of  $w_{ki}$  is determined according to the absolute error between the observed values and the simulated values. The weights of different models at different points are variational, and the point with smaller absolute error will be given bigger weight. The weight can be calculated as:

$$w_{ki} = \frac{1/|e_{ki}|}{\sum_{k=1}^K 1/|e_{ki}|} \quad (21)$$

where  $|e_{ki}|$  denotes the absolute error of the  $k$ th model at point  $i$ .

#### 2.4.3. Artificial Neural Network-Based Ensemble

Because the spread of the GRNN model is the only parameter to be optimized, error caused by uncertainty in the parameters can be decreased. Thus, the GRNN network was chosen as the ensemble technique to enhance the performance of the single ANN. In the GNE model, the forecasting results of the RBF, ELM, and Elman networks were taken as input variables, while the target variable was taken as the output variable. The GRNN network was trained to obtain the best model parameter. The forecasting result of the GRNN network was taken as the result of the GNE model. The weight coefficients of the RBF, ELM, and Elman models were variational and the accurate weight of these models could not be obtained because of the black-box principle of the GRNN neural network. The structure of the GRNN ensemble forecasting model, using the RBF, ELM, and Elman models, is shown in Figure 1.



**Figure 1.** The structure of the GNE model based on RRF, ELM and Elman networks.

## 2.5. Model Performance Evaluation

It is important to apply multiple error measure indices when evaluating the forecasting ability of the developed models. Four measures, the root mean square error (RMSE), the mean absolute error (MAE), the mean absolute percent error (MAPE), and the coefficient of correlation ( $R_c$ ) have been used in this paper [35]. Among the four statistical measures, RMSE was sensitive to the extremely large or small values of a time series and reflected the degree of variation, the MAE reflected the actual forecasting error in a more balanced perspective, and the MAPE was a measure of accuracy for the forecasted streamflow series with no units. The RMSE, MAE, and MAPE are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (q_p(i) - q_o(i))^2} \quad (22)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |q_p(i) - q_o(i)| \quad (23)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|q_p(i) - q_o(i)|}{q_o(i)} \times 100\% \quad (24)$$

where  $q_p(i)$  and  $q_o(i)$  are the predicted and observed monthly streamflow series, respectively, and  $N$  is the length of the data.

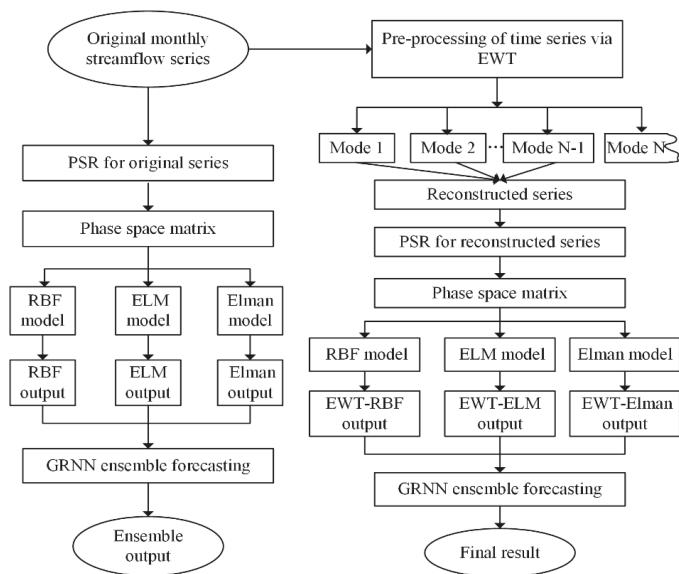
The  $R_c$  describes the degree to which two data sets are related and ranges from  $-1$  to  $1$ . The larger the absolute value of the correlation coefficient, the more the predicted and observed data is related. The correlation coefficient is defined as:

$$R_c = \frac{(1/N) \sum_{i=1}^N (q_p(i) - \bar{q}_p)(q_o(i) - \bar{q}_o)}{\sqrt{(1/N) \sum_{i=1}^N (q_p(i) - \bar{q}_p)^2} \times \sqrt{(1/N) \sum_{i=1}^N (q_o(i) - \bar{q}_o)^2}} \quad (25)$$

where  $\bar{q}_p$  and  $\bar{q}_o$  indicate the average value of the predicted and observed runoffs, respectively.

## 2.6. Modeling Framework

Figure 2 illustrates the detailed procedures of schematic structures of the various models used in this study. The proposed EWT-GNE model could be implemented in four steps. The first step was the denoising process of the original streamflow time series. In this step, the streamflow series was divided into four independent modes using the EWT algorithm. The mode with the highest frequency was removed to eliminate redundant noise. In the second step, the PSR method was used to construct the phase space matrix, namely, the input matrix of the neural networks. In the third step, three individual ANN models, RBF, ELM, and Elman networks, were used to forecast the monthly streamflow time series independently. The three varying architectures of ANNs were chosen because they are three of the most representative ANNs that have been used for forecasting. The RBF network is one of the most widely used FNNs. The ELM network is a new type of SLFNs in which weights and biases are randomly assigned. The Elman network is a type of dynamic recurrent neural network. The last step was to group the results of the three individual models using GRNN. In this part, the forecasted outputs of the RBF, ELM, and Elman networks were taken as input variables of the GNE model. The forecasting result of the GRNN model was the result of the proposed EWT-GNE model. The GRNN network was chosen as an ensemble technique because it has only one parameter to be optimized. Accordingly, errors caused by parameter uncertainty can be decreased. The left portion of Figure 2 illustrates the modelling framework of the models without the EWT denoising technique, while the right portion of Figure 2 illustrates that of the models with the EWT denoising technique.



**Figure 2.** A schematic view of the modeling framework.

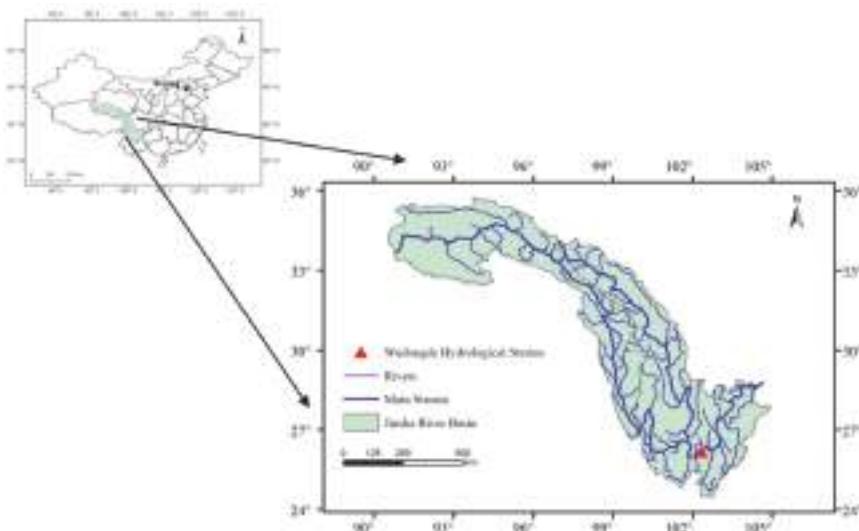
### 3. Model Construction and Development

#### 3.1. Study Area and Data Collection

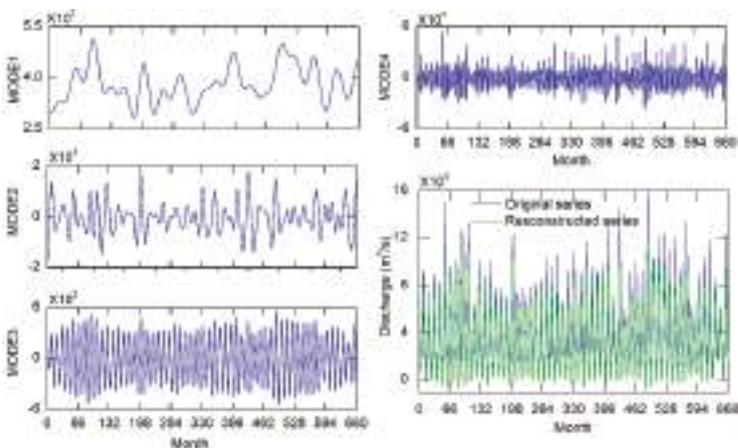
The Jinsha River is located in the upper portion of the Yangtze River and originates in the Tanggula Mountains. It flows through Sichuan, Yunan and Tibet provinces in China. The upper portion of the Yangtze River located in Zhimenda, Yushu in Qinghai Province is called the Jinsha River, the length of which is 2326 km and the height difference is 3280 km. The catchment area of Jinsha River basin is 473,000 km<sup>2</sup>. The Jinsha River Basin can be subdivided into upper, middle, and lower sections. The monthly streamflow data of the Wudongde hydropower station, which is located in the lower portion of the Jinsha River basin, is studied in this paper. The catchment area of the Wudongde hydropower station is 406,100 km<sup>2</sup>, which accounts for 86% of the total area of the Jinsha River basin. The average annual discharge is 3810 m<sup>3</sup>/s and the total runoff is 1200 billion m<sup>3</sup>. The observed data ranged from January 1958 to December 2012, with a total length of 55 years (660 months). The first 525 months of the streamflow data (January 1958 to September 2001) were selected for training. The remaining 135 months (October 2001 to December 2012) were used for validation. Figure 3 illustrates the locations of the Jinsha River basin and the Wudongde hydrological station.

#### 3.2. Data Preprocessing Using Empirical Wavelet Transform

Before submitting the streamflow series into the forecasting model, the original series was preprocessed using EWT. The original streamflow series was divided into four independent modes. The mode with the highest frequency was discarded. A visual representation of the decomposed subseries and the comparison between the original and denoised series is shown in Figure 4. Following preprocessing, the peaks of the original series were weakened and the troughs were lowered. The oscillations of the monthly streamflow time series were eliminated to a certain extent using EWT.



**Figure 3.** Locations of the Jinsha River basin and the Wudongde hydrological station.

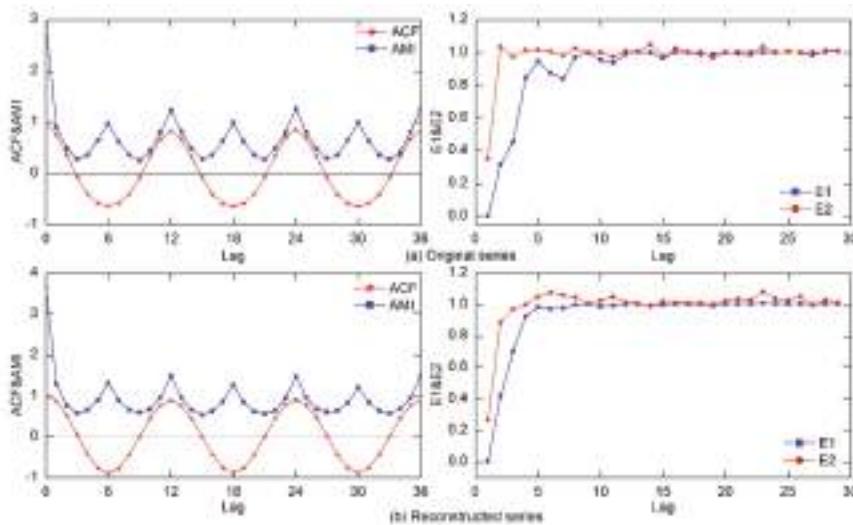


**Figure 4.** Visual representation of the decomposed subseries and the comparison between the original and reconstructed series.

### 3.3. Determination of Phase Space Reconstruction Parameters

After the denoising process was completed, the one-dimensional denoised streamflow series was reconstructed into a multi-dimensional phase space matrix for forecasting. To determine the delay time  $\tau$  of the PSR method, the ACF and AMI values of the original and reconstructed series were calculated. As can be seen in Figure 5, for both the original and the reconstructed series, the AMI values reached the first minimum at lag 3; the ACF values also attained zero. Because the ACF and AMI values gave the same determination of delay time  $\tau$ ,  $\tau$  was chosen as 3 for both the original and the reconstructed time series. After determining the delay time, the Cao method was used to determine the embedded dimension  $m$ . As can be seen in Figure 5, because  $E_1(m)$  almost stopped changing when  $m$  was greater than 11,  $m$  was chosen as 12 for the original time series, while 8 was

chosen for the reconstructed series. After the determination of  $\tau$  and  $m$ , the maximum Lyapunov exponent  $\lambda_m$  was computed for both series.  $\lambda_m$  was calculated as 2.263 and 4.142 for the original series and the reconstructed series, respectively. Since the largest Lyapunov exponent was determined to be positive, the streamflow series was identified as chaotic. Additionally, the maximum Lyapunov exponent of the original series was much larger than the reconstructed series, which demonstrated that the reconstructed streamflow time series was less disordered and unsystematic than the original time series, according to chaos theory. Discarding the mode with the highest frequency eliminated the redundant noises of the original monthly streamflow time series. Subsequently, the EWT method could be used as a data preanalysis technique in the forecasting of the monthly streamflow series in this study.



**Figure 5.** The auto-correction function (ACF) & Average Mutual Information (AMI) values and Cao method results plot for the original and reconstructed streamflow series.

### 3.4. Parameter Settings of Different ANNs

The diversity of the submodels was realized through the use of various ANN architectures. For the three submodels, the total number of hidden neurons was determined using a grid search (GS) algorithm to assure fair and valid comparisons. The search range was set as  $[m, 2n + 20]$ , where  $n$  denotes the number of input neurons,  $m$  is set as  $2n - 20$  if  $n$  is bigger than 10; otherwise  $m$  was set at 1. The searching step was set at 1. The transfer functions of hidden and output layers were of sigmoid and linear types for the Elman neural networks, respectively, while that of the hidden layer of the ELM network was linear. With regard to the RBF neural network, the spreads ranged from 0.1 to 4.0, with 0.1 increment to obtain the best forecasting performance. The spread was the only parameter to be optimized for the GRNN network. The spreads ranged from 0.02 to 0.1, with 0.01 increment. The optimal parameters for neural networks used in this study for streamflow forecasting are shown in Table 1.

**Table 1.** Optimal parameters of the ANN models used for streamflow forecasting.

Models	RBF	EWT-RBF	ELM	EWT-ELM	Elman	EWT-Elman	GNE	EWT-GNE
Neurons	20	20	43	35	9	20	—	—
Spread	1.4	1.8	—	—	—	—	0.045	0.029

#### 4. Results and Analysis

The comparison and analysis of results can be divided into four steps. First, the comparisons between the methods with and without EWT were conducted to demonstrate the effectiveness of the EWT-based preprocessing method in increasing the accuracy of streamflow time series forecasting. Second, the performance of the three ANN models, e.g., the RBF, Elman, and ELM models, were analyzed to investigate their ability in monthly streamflow forecasting as well as to recommend the most appropriate model. Thirdly, comparisons among the predictions of the various individual and ensemble models were investigated to verify the efficiency of the ensemble techniques. Finally, the performance of the various ensemble techniques, including SAE, WAE, and GNE, were compared to highlight the effectiveness of the GRNN-based ensemble technique.

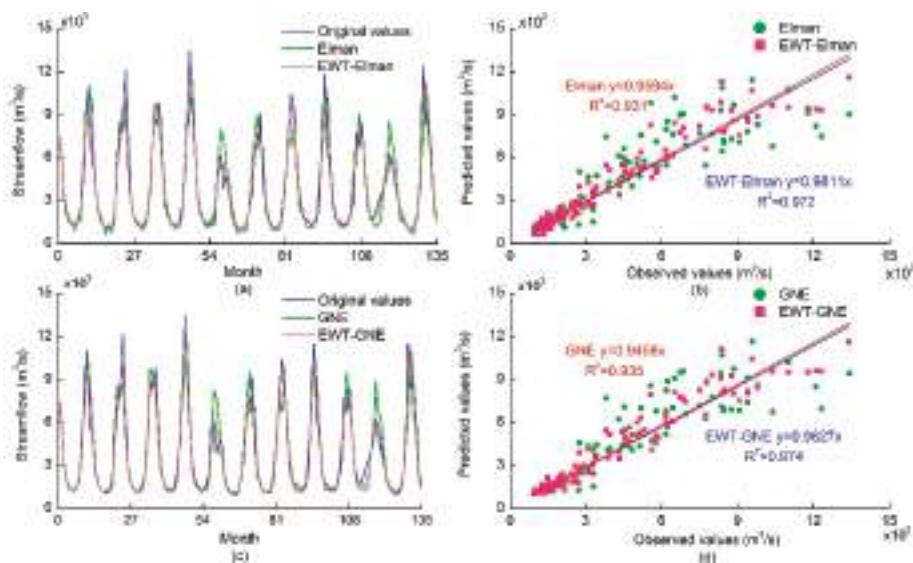
The performance evaluation indices of the 12 models developed in this study, including RBF, ELM, Elman, SAE, WAE, and GNE model, with and without the EWT algorithm, in the training and validation periods are shown in Tables 2 and 3, respectively. It can be concluded from the preliminary analysis of results between the non-denoising models and the EWT-based models that the performances of the latter were superior to that of the former in terms of the four performance indices. It can be seen from Tables 2 and 3 that the EWT-based models performed much better than the corresponding non-denoising models. The RMSE, MAE, and MAPE values of the former were all smaller than the latter, while the R value was significantly larger. To display the efficiency of the EWT-based denoising technique, the model performance of the Elman and GNE models with and without EWT in the validation period are shown in Figure 6, where the solid lines on the left represent both the predicted and the observed values, while the right shows the scatter plots. It can be seen from Figure 6 that the EWT-based models (EWT-Elman and EWT-GNE) approximated the original streamflow time series better than the corresponding non-denoising models (Elman and GNE), and the scatters distributed more tightly around the least-square regression line.

**Table 2.** The forecasting results of the models in the training period.

Models	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	R	MAPE (%)	Models	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	R	MAPE (%)
RBF	1175.70	710.11	0.931	17.751	EWT-RBF	748.72	480.94	0.973	13.508
ELM	1138.34	711.62	0.935	19.445	EWT-ELM	712.07	462.07	0.975	13.046
Elman	1071.81	641.72	0.943	15.380	EWT-Elman	715.71	463.51	0.975	13.551
SAE	1103.51	648.95	0.939	15.513	EWT-SAE	711.13	443.49	0.975	11.362
WAE	1070.36	574.04	0.943	12.268	EWT-WAE	693.70	399.46	0.976	9.204
GNE	920.53	548.81	0.958	13.055	EWT-GNE	613.16	385.95	0.982	10.237

**Table 3.** The forecasting results of the models in the validation period.

Models	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	R	MAPE (%)	Models	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	R	MAPE (%)
RBF	1351.04	796.64	0.898	20.342	EWT-RBF	854.68	574.94	0.959	17.446
ELM	1308.82	796.89	0.903	19.350	EWT-ELM	833.07	554.03	0.961	15.679
Elman	1292.60	796.78	0.906	20.045	EWT-Elman	821.11	543.21	0.962	14.455
SAE	1293.75	752.28	0.906	17.159	EWT-SAE	820.40	538.90	0.962	13.699
WAE	1264.56	677.82	0.910	14.016	EWT-WAE	796.13	481.35	0.965	11.410
GNE	1246.47	753.16	0.913	17.602	EWT-GNE	790.35	527.15	0.965	13.541



**Figure 6.** Forecasting results of the Elman (GNE) and EWT-Elman (EWT-GNE) models in the validation stage.

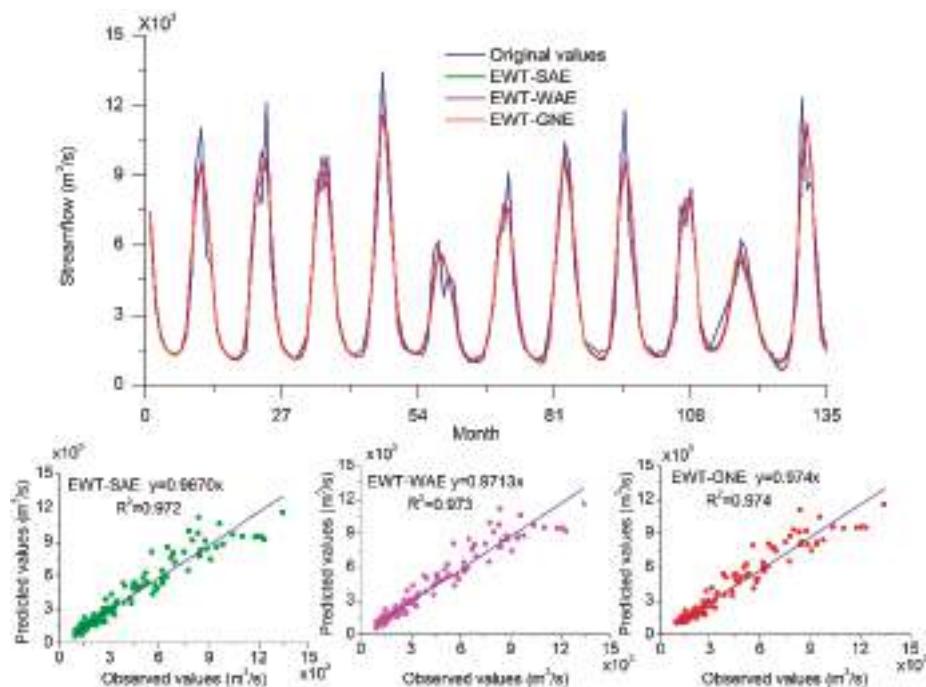
To further demonstrate variations in the forecasting performance between the non-denoising models and the EWT-based models, the improved percentages between the two kinds of methods were calculated. The RMSE, MAE, R, and MAPE differences within the EWT-Elman model and the Elman model were 33.22%, 27.77%, 3.39%, 11.89% in the training stage and 36.48%, 31.82%, 3.39%, 27.89% in the validation stage, respectively. For the EWT-GNE model and the GNE models, the RMSE, MAE, R, and MAPE differences were 35.55%, 32.27%, 2.75%, 24.16% and 36.56%, 29.99%, 5.77%, 22.93% in the training and validation stages, respectively. The comparisons between the EWT-based models and the non-denoising models have demonstrated that the EWT denoising process is effective in improving the prediction accuracy of the monthly streamflow.

From the results of Tables 2 and 3, it can also be seen that the Elman model performed the best in comparison with the RBF and ELM models; additionally, the performance of the RBF model was the worst. The differences between ELM and Elman models were not significant with respect to RMSE, MAE, MAPE, and R criteria. The RMSE, MAE, MAPE, and R of the Elman model consistently performed either equal to or better than the ELM model. The improvements of the RMSE, MAE, MAPE, and R of the Elman model to ELM model were 1.44%, 1.95%, -0.10%, and 7.81% in the validation stage, respectively.

It can also be seen in Tables 2 and 3 that all ensemble models performed better than the individual models. The ensemble models (GNE and EWT-GNE) performed better than the corresponding best-performing individual model (Elman and EWT-Elman). The RMSE, MAE, R, and MAPE differences between the GNE and Elman models were 11.23%, 11.20%, 1.31%, 12.24% in the training stage and 3.62%, 5.49%, 0.74%, 12.35% in the validation stage. In contrast, those between the EWT-GNE and EWT-Elman models were 14.33%, 16.73%, 0.68%, 24.46% in the training stage and 3.75%, 2.96%, 0.35%, 6.32% in the validation stage, which demonstrated the effectiveness of the proposed GRNN-ensemble technique in enhancing the predicting ability of ANN models.

From a detailed comparison of the different ensemble models with EWT, according to Tables 2 and 3, it can be seen that the four indices of both the EWT-WAE and EWT-GNE models were superior to those of the EWT-SAE model. The EWT-GNE model performed slightly better than the EWT-WAE model. The improvement of RMSE, MAE, R, and MAPE was 11.61%, 3.38%, 0.58%,

−11.22% in the training stage and 0.73%, −9.52%, 0.03%, −18.68% in the validation stage between EWT-WAE and EWT-GNE models. Additionally, to display the differences between the various ensemble models, the forecasting results of the EWT-SAE, EWT-WAE, and EWT-GNE models are shown in Figure 7. From the illustrations of Figure 7, it can be seen that the predicted values of the EWT-GNE approximated the observed values better than the EWT-SAE model, with the scatters distributed more tightly around the regression line. Thus, the superiority of the proposed GRNN-based ensemble technique over the other ensemble techniques was demonstrated.



**Figure 7.** Forecasting results of the simple averaging ensemble (SAE), weighted averaging ensemble (WAE) and GRNN-based ensemble (GNE) models with EWT in the validation stage.

## 5. Conclusions

The present study developed and tested an ANN ensemble model for monthly streamflow forecasting through an application to a case study in China. Three neural network architectures (RBF, ELM, and Elman) were used as sub-ANNs for forecasting. Results demonstrate that all three ANNs performed well, with the best performance achieved by the Elman network. To improve the generalizability and prediction accuracy of the ANNs, various ensemble techniques as the SAE, the WAE, and the GNE were proposed to combine the outputs of the sub-ANNs. The ensemble models achieved better prediction results compared with individual models, with the GRNN-ensemble model having performed the best. As a result of the volatility of the monthly streamflow time series, EWT was used to filter noise. The denoised streamflow time series was less disordered and unsystematic than the original time series according to chaos theory. The models with the EWT-based denoising process outperformed the non-denoising ones. Overall, results show that the proposed EWT-GNE model can be used as a successful tool for monthly streamflow forecasting. The proposed GRNN-ensemble technique can decrease the unpredictability of single ANN forecasting models, and the EWT algorithm can filter the noises of the streamflow series, providing more accurate forecasting results.

In the future, one can apply the techniques developed in this study to streamflow data in different time scales and from other hydrological stations. Hydro-meteorological data such as rainfall [8,14], precipitation, and streamflow data from adjacent hydrological stations [30] can be considered as model inputs to enhance streamflow forecasting accuracy. Additionally, the newest machine learning techniques, such as deep learning [36], can be considered in terms of streamflow forecasting.

**Author Contributions:** J.Z. and T.P. designed the experiments; T.P. wrote the draft of the paper; C.Z. prepared the figures; J.Z. and N.S. provided useful advices and made some corrections. All authors read and approved the final manuscript.

**Funding:** This work is supported by the Key Program of the Major Research Plan of the National Natural Science Foundation of China (No. 91547208), the National Natural Science Foundation of China (No. 51579107), and the National Key R&D Program of China (2016YFC0402708, 2016YFC0401005).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Hong, M.; Wang, D.; Wang, Y.; Zeng, X.; Ge, S.; Yan, H.; Singh, V.P. Mid- and long-term runoff predictions by an improved phase-space reconstruction model. *Environ. Res.* **2016**, *148*, 560–573. [[CrossRef](#)] [[PubMed](#)]
- Peng, T.; Zhou, J.; Zhang, C.; Fu, W. Streamflow forecasting using empirical wavelet transform and artificial neural networks. *Water* **2017**, *9*, 406. [[CrossRef](#)]
- Hong, W.C.; Pai, P.F. Potential assessment of the support vector regression technique in rainfall forecasting. *Water Resour. Manag.* **2007**, *21*, 495–513. [[CrossRef](#)]
- Solomatine, D.P.; Ostfeld, A. Data-driven modelling: Some past experiences and new approaches. *J. Hydroinform.* **2008**, *10*, 3–22. [[CrossRef](#)]
- Pumo, D.; Viola, F.; Noto, L.V. Generation of natural runoff monthly series at ungauged sites using a regional regressive model. *Water* **2016**, *8*, 209. [[CrossRef](#)]
- Pumo, D.; Conti, F.L.; Viola, F.; Noto, L.V. An automatic tool for reconstructing monthly time-series of hydro-climatic variables at ungauged basins. *Environ. Model. Softw.* **2017**, *95*, 381–400. [[CrossRef](#)]
- Zhang, C.; Zhou, J.; Li, C.; Fu, W.; Peng, T. A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting. *Energy Convers. Manag.* **2017**, *143*, 360–376. [[CrossRef](#)]
- Chen, L.; Singh, V.P.; Guo, S.; Zhou, J.; Ye, L. Copula entropy coupled with artificial neural network for rainfall-runoff simulation. *Stoch. Environ. Res. Risk Assess.* **2014**, *28*, 1755–1767. [[CrossRef](#)]
- Chang, F.J.; Chen, P.A.; Lu, Y.R.; Huang, E.; Chang, K.Y. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* **2014**, *517*, 836–846. [[CrossRef](#)]
- Hosseini-Moghari, S.M.; Araghinejad, S. Monthly and seasonal drought forecasting using statistical neural networks. *Environ. Earth Sci.* **2015**, *74*, 397–412. [[CrossRef](#)]
- Zhou, T.; Wang, F.; Yang, Z. Comparative analysis of ANN and SVM models combined with wavelet preprocess for groundwater depth prediction. *Water* **2017**, *9*, 781. [[CrossRef](#)]
- Yu, Y.; Zhang, H.; Singh, V. Forward prediction of runoff data in data-scarce basins with an improved ensemble empirical mode decomposition (EEMD) model. *Water* **2018**, *10*, 388. [[CrossRef](#)]
- Wang, W.C.; Chau, K.W.; Qiu, L.; Chen, Y.B. Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on EEMD decomposition. *Environ. Res.* **2015**, *139*, 46–54. [[CrossRef](#)] [[PubMed](#)]
- Zhu, S.; Zhou, J.; Ye, L.; Meng, C. Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze river, China. *Environ. Earth Sci.* **2016**, *75*, 531. [[CrossRef](#)]
- Seo, Y.; Kim, S.; Kisi, O.; Singh, V.P. Daily water level forecasting using wavelet decomposition and artificial intelligence techniques. *J. Hydrol.* **2015**, *520*, 224–243. [[CrossRef](#)]
- Gilles, J. Empirical wavelet transform. *IEEE Trans. Signal Process.* **2013**, *61*, 3999–4010. [[CrossRef](#)]
- Hu, J.; Wang, J. Short-term wind speed prediction using empirical wavelet transform and Gaussian process regression. *Energy* **2015**, *93*, 1456–1466. [[CrossRef](#)]

18. Wang, J.; Hu, J. A robust combination approach for short-term wind speed forecasting and analysis—Combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model. *Energy* **2015**, *93*, 41–56.
19. Zhao, X.; Chen, X.; Xu, Y.; Xi, D.; Zhang, Y.; Zheng, X. An EMD-based chaotic least squares support vector machine hybrid model for annual runoff forecasting. *Water* **2017**, *9*, 153. [[CrossRef](#)]
20. Guo, J.; Zhou, J.; Qin, H.; Zou, Q.; Li, Q. Monthly streamflow forecasting based on improved support vector machine model. *Expert Syst. Appl.* **2011**, *38*, 13073–13081. [[CrossRef](#)]
21. Hu, Z.; Zhang, C.; Luo, G.; Teng, Z.; Jia, C. Characterizing cross-scale chaotic behaviors of the runoff time series in an inland river of Central Asia. *Quat. Int.* **2013**, *311*, 132–139. [[CrossRef](#)]
22. Ouyang, Q.; Lu, W.; Xin, X.; Zhang, Y.; Cheng, W.; Yu, T. Monthly rainfall forecasting using EEMD-SVR based on phase-space reconstruction. *Water Resour. Manag.* **2016**, *30*, 2311–2325. [[CrossRef](#)]
23. Pumo, D.; Francipane, A.; Conti, F.L.; Arnone, E.; Bitonto, P.; Viola, F.; La Loggia, G.; Noto, L.V. The SESAMO early warning system for rainfall-triggered landslides. *J. Hydroinform.* **2016**, *18*, 256–276. [[CrossRef](#)]
24. Ren, Y.; Suganthan, P.N.; Srikanth, N. Ensemble methods for wind and solar power forecasting—A state-of-the-art review. *Renew. Sustain. Energy Rev.* **2015**, *50*, 82–91. [[CrossRef](#)]
25. Jovanović, R.Ž.; Sretenović, A.A.; Živković, B.D. Ensemble of various neural networks for prediction of heating energy consumption. *Energy Build.* **2015**, *94*, 189–199. [[CrossRef](#)]
26. Lin, G.F.; Chen, L.H. A non-linear rainfall-runoff model using radial basis function network. *J. Hydrol.* **2004**, *289*, 1–8. [[CrossRef](#)]
27. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 982, pp. 985–990.
28. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
29. Specht, D.F. A general regression neural network. *IEEE Trans. Neural Netw.* **1991**, *2*, 568–576. [[CrossRef](#)] [[PubMed](#)]
30. Chen, L.; Ye, L.; Singh, V.; Asce, F.; Zhou, J.; Guo, S. Determination of input for artificial neural networks for flood forecasting using the copula entropy method. *J. Hydrol. Eng.* **2014**, *19*, 217–226. [[CrossRef](#)]
31. Packard, N.H.; Crutchfield, J.P.; Farmer, J.D.; Shaw, R.S. Geometry from a time series. *Phys. Rev. Lett.* **1980**, *45*, 712. [[CrossRef](#)]
32. Cao, L. Practical method for determining the minimum embedding dimension of a scalar time series. *Phys. D Nonlinear Phenom.* **1997**, *110*, 43–50. [[CrossRef](#)]
33. Wu, C.L.; Chau, K.W.; Fan, C. Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. *J. Hydrol.* **2010**, *389*, 146–167. [[CrossRef](#)]
34. Rosenstein, M.T.; Collins, J.J.; Luca, C.J.D. A practical method for calculating largest Lyapunov exponents from small data sets. *Phys. D Nonlinear Phenom.* **1993**, *65*, 117–134. [[CrossRef](#)]
35. Peng, T.; Zhou, J.; Zhang, C.; Zheng, Y. Multi-step ahead wind speed forecasting using a hybrid model based on two-stage decomposition technique and AdaBoost-extreme learning machine. *Energy Convers. Manag.* **2017**, *153*, 589–602. [[CrossRef](#)]
36. Bai, Y.; Chen, Z.; Xie, J.; Li, C. Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models. *J. Hydrol.* **2016**, *532*, 193–206. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Physical Hybrid Neural Network Model to Forecast Typhoon Floods

You-Da Jhong <sup>1</sup>, Chang-Shian Chen <sup>2</sup>, Hsin-Ping Lin <sup>2</sup> and Shien-Tsung Chen <sup>2,\*</sup>

<sup>1</sup> Construction and Disaster Prevention Research Center, Feng Chia University, Taichung 407, Taiwan; ydjhong@fcu.edu.tw

<sup>2</sup> Department of Water Resources Engineering and Conservation, Feng Chia University, Taichung 407, Taiwan; cschen@fcu.edu.tw (C.-S.C.); zxp86285@yahoo.com.tw (H.-P.L.)

\* Correspondence: stchen@fcu.edu.tw; Tel.: +886-4-2451-7250

Received: 10 April 2018; Accepted: 10 May 2018; Published: 13 May 2018

**Abstract:** This study proposed a hybrid neural network model that combines a self-organizing map (SOM) and back-propagation neural networks (BPNNs) to model the rainfall-runoff process in a physically interpretable manner and to accurately forecast typhoon floods. The SOM and a two-stage clustering scheme were applied to group hydrologic data into four clusters, each of which represented a meaningful hydrologic component of the rainfall-runoff process. BPNNs were constructed for each cluster to achieve high forecasting capability. The physical hybrid neural network model was used to forecast typhoon flood discharges in Wu River in Taiwan by using two types of rainfall data. The clustering results demonstrated that the rainfall-runoff process was favorably described by the sequence of derived clusters. The flood forecasting results indicated that the proposed hybrid neural network model has good forecasting capability, and the performance of the models using the two types of rainfall data is similar. In addition, the derived lagged inputs are hydrologically meaningful, and the number and activation function of the hidden nodes can be rationally interpreted. This study also developed a traditional, single BPNN model trained using the whole calibration data for comparison with the hybrid neural network model. The proposed physical hybrid neural network model outperformed the traditional neural network model in forecasting the peak discharges and low flows.

**Keywords:** hybrid neural network; flood forecasting; self-organizing map

## 1. Introduction

Flood forecasting is an important nonstructural measure for flood mitigation during flash floods caused by typhoons. Machine learning methods have been widely employed to develop flood forecasting models. Among them, the artificial neural network (ANN) and its hybrids are competent techniques widely used for flood forecasting. For example, ANNs have been successfully employed to forecast rainfall and flood for decades [1–7]. Support vector machines (SVMs), which have the same network architecture as the radial basis function neural network [8], have recently gained popularity and exhibited good performance in rainfall and flood forecasting [9–17]. In addition, the neurofuzzy system, which combines the ANN and fuzzy inference system, has been favorably applied in various hydrologic forecasting studies [18–25].

Although ANN-based models exhibit high forecasting performance, they are regarded as a black box and their physical interpretation is unattainable. For instance, Zhang et al. [26] and Lange [27] noted that the ANN has no explicit form for analyzing the relationship between inputs and outputs and that explaining the results obtained by the networks is difficult. Some studies have attempted to interpret the physical meaning of the derived network structure and results of ANN models. Jain et al. [28] demonstrated that the hidden neurons in the ANN rainfall-runoff model approximate

various components of the hydrologic process, such as infiltration, base flow, and surface flow. Chen and Yu [12] and Chen [29] demonstrated that the input data mined to construct the hidden nodes of an SVM network are informative hydrologic data that characterize a flood hydrograph, particularly the data around the peak flood and in the rising limb.

Separate ANN models trained using different input–output data sets have been proposed to improve forecasting performance. For example, Furundzic [30] used the self-organizing map (SOM) to decompose input–output data into three sets and develop separate ANNs for each data set. Abrahart and See [31], Hsu et al. [32], and Jain and Srinivasulu [33] also applied the SOM to partition data into different clusters corresponding to the different segments of the hydrograph and developed separate ANN models for each cluster. They concluded that the performance of the separate ANNs is better than that of a single ANN trained using the whole dataset. The partitioning of data is based on the fact that different magnitudes of hydrologic data are produced by different physical processes. A separate ANN can more closely model a specific dataset corresponding to a hydrologic component. However, most studies that applied data partitioning focused on the decomposition method and performance improvement. Efforts in physically partitioning data and interpreting the hydrologic process have been limited.

The present study, which also partitioned data into clusters and constructed separate ANNs, focused on the derivation of physically interpretable clusters. The SOM was applied to group hydrologic data into four clusters. The sequence of these clusters physically represents the rainfall-runoff process of a storm event according to the quantity of the rainfall and discharge data. A two-stage clustering scheme was used to obtain the expected meaningful clusters. A back-propagation neural network (BPNN) was employed to construct the forecasting model for each cluster to forecast flood discharge. The proposed hybrid neural network model that combines the SOM and BPNN characterizes the rainfall-runoff process in a physically interpretable manner. The physical hybrid neural network model was used to forecast typhoon flood discharges in Wu River in Taiwan. Two types of forecasting model were constructed with respect to two sets of rainfall data (the basin average rainfall and rainfall from different rain gauges). The clustering results prove that the proposed clustering scheme captures the behavior of the rainfall-runoff process and properly divides the hydrologic process into different components. Flood forecasting results reveal that both types of forecasting models have favorable forecasting capability with high coefficient of efficiency values and low mean absolute errors. In addition, the proposed hybrid neural network model was compared with a single traditional neural network model that was constructed using the whole dataset. The following section introduces the proposed physical hybrid neural network model including the methodologies of SOM and BPNN. Section 3 provides information of the study area and typhoon flood data. Section 4 presents the model development process and the flood forecasting performed by the hybrid neural network model. A comparison of the proposed model to the traditional BPNN model is presented as well. The last section outlines the conclusions of this study.

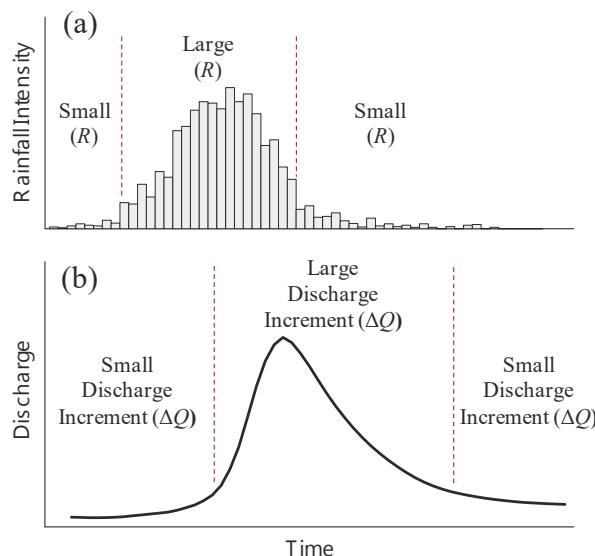
## 2. Physical Hybrid Neural Network Model

### 2.1. Rainfall-Runoff Clusters Based on the Hydrologic Process

The rainfall-runoff process can be divided into several temporal steps corresponding to respective hydrologic phenomena. This study grouped rainfall-runoff data into clusters to represent different components of the rainfall-runoff process. An example of a typical rainfall event is shown in Figure 1a. Such an event generally begins with low rainfall ( $R$ ), followed by intense, heavy rainfall, and finally ends with sprinkling rainfall. A storm hydrograph recorded during a storm rainfall event is presented in Figure 1b. At the beginning of the event, the discharge ( $Q$ ) rises slowly and the discharge increment ( $\Delta Q$ ) is small. As the high intensity rainfall continues, the discharge increases rapidly to the peak discharge during the rising limb of the hydrograph. In the major part of the rising limb, the discharge increment is large. After cessation of the intense rainfall, the discharge declines sharply, but the

discharge increment remains large. In the lower part of the recession limb, the discharge decreases slowly to the base flow and the discharge increment is small.

On the basis of the typical rainfall-runoff event illustrated in Figure 1, the rainfall-runoff process can be divided into several steps, as shown in Figure 2. At the beginning of a rainfall event, the rainfall is generally low and does not significantly contribute to the runoff. The rainfall-runoff data during this step are low rainfall and small discharge increments, and the data are grouped as Cluster A. Next, the rainfall increases and becomes large; however, the initial losses and high infiltration losses during this period cause only a gradual increase in the discharge. The high rainfall and small discharge increment data during this period are grouped as Cluster B. Subsequently, the infiltration losses decrease and increasingly more surface runoff reaches the basin outlet. The discharge increases rapidly to the crest segment of the hydrograph. The high rainfall and large discharge increment data are grouped as Cluster C. When the rainfall diminishes and the discharge starts to decrease, the data with low rainfall and large discharge increment are grouped as Cluster D. Subsequently, the segment of a hydrograph with low rainfall and small discharge increment is similar to the initial part of the hydrograph, and the corresponding data are thus also grouped as Cluster A. Thus, the rainfall-runoff process can be described by the sequence through Clusters A, B, C, D, and A. For an actual storm hydrograph, the rainfall-runoff phenomenon can be more complex, and the translation and storage effects can be significant in large watersheds. However, the proposed four clusters represent the rainfall-runoff process for a typical storm hydrograph.

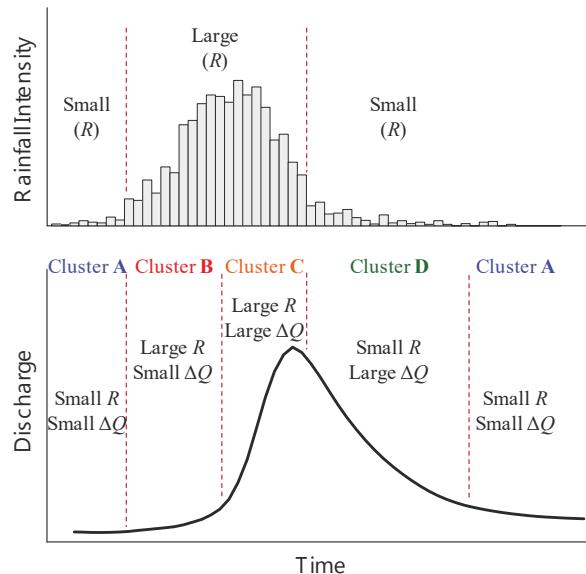


**Figure 1.** Example of a typical rainfall and runoff event. (a) Rainfall hyetograph; (b) Flood hydrograph.

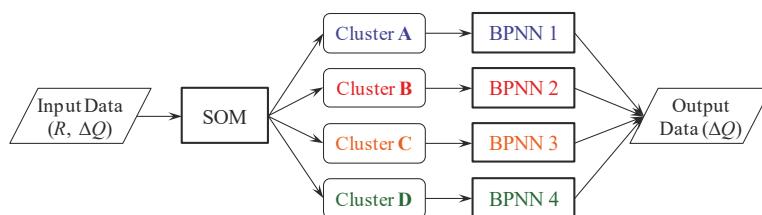
## 2.2. Hybrid Neural Network Model

The proposed hybrid neural network model based on physically clustered hydrologic data is illustrated in Figure 3. The input rainfall and discharge increment data are grouped into four clusters using the SOM. Each cluster meaningfully corresponds to a typical step in the rainfall-runoff process. Then, BPNNs are constructed with respect to each cluster to forecast the discharge increment. The discharge forecasts are obtained when the forecasted discharge increment is added to the observed discharge at the present time. The detailed methodology of the SOM and BPNN has been well

documented in the literature. Therefore, a brief description of the two neural network models is provided herein.



**Figure 2.** Rainfall-runoff clusters based on the hydrologic process.



**Figure 3.** Structure of the proposed hybrid neural network model.

### 2.2.1. SOM

The SOM, proposed by Kohonen [34], is an unsupervised-learning neural network that automatically groups input data into several clusters without assigning the target outputs. The SOM uses a competitive learning strategy to map the input data onto a low-dimensional topological map. The process of constructing an SOM neural network is described briefly as follows.

The SOM network comprises one input layer and one output layer (the topological map), and the input neurons are fully connected to the output neurons. Let the input variables  $x_i$  ( $i = 1, 2, \dots, m$ ) form an input vector  $\mathbf{X}$ , where  $m$  is the number of input neurons. Each output neuron  $u_j$  ( $j = 1, 2, \dots, n$ ) on the topological map has a weight  $w_{ij}$  with respect to each input variable  $x_i$ , and  $n$  is the number of output neurons. The SOM is trained iteratively using randomly assigned initial weights. The SOM algorithm calculates the similarity between the input vector  $\mathbf{X}$  and weight vector  $\mathbf{W}_j$  for each output neuron. The similarity is defined as the Euclidean distance  $d_j$ :

$$d_j = \|\mathbf{X} - \mathbf{W}_j\| = \left( \sum_{i=1}^m (x_i - w_{ij})^2 \right)^{1/2} \quad (1)$$

The output neuron whose weight vector is closest to the input vector has the minimum distance and is declared the winning neuron. The weights of this winning neuron  $u_j^*$  and its neighboring neurons  $u_j$  are then adjusted to approach the input vector. A typical neighborhood function is the Gaussian function  $h_j$ :

$$h_j = \exp \left( -\frac{\|u_j - u_j^*\|^2}{2\sigma^2} \right) \quad (2)$$

where  $\sigma$  is the width of the topological neighborhood. The neighborhood function  $h_j$  and width  $\sigma$  are usually set to decrease monotonically during the iterative process. The adjusted weight at iteration time  $r + 1$  is defined as

$$\mathbf{W}_j(r+1) = \mathbf{W}_j(r) + \eta(r) \cdot h_j(r) \cdot [\mathbf{X} - \mathbf{W}_j(r)] \quad (3)$$

where  $\eta$  is the learning rate ( $0 < \eta < 1$ ) and is also set to decrease during the iterative process. Iterations are performed until the weight vector converges. Thereafter, similar input vectors are mapped to a specific region (cluster) on the topological map, and several clusters are automatically grouped.

## 2.2.2. BPNN

The BPNN, developed by Rumelhart et al. [35], is the most representative and popularly used neural network. A supervised multilayer feed-forward neural network, the BPNN uses the back-propagation algorithm for network training. The BPNN typically comprises three layers: the input, hidden, and output layers. Let the input variables  $x_i$  ( $i = 1, 2, \dots, m$ ) be the neurons in the input layer, and  $\hat{y}_k$  ( $k = 1, 2, \dots, p$ ) be the output variable of the  $k$ -th neuron in the output layer. The BPNN output  $\hat{y}_k$  is expected to fit the target (actual) output  $y_k$ . The BPNN (with  $n$  neurons in the hidden layer) can be expressed in the following form:

$$\hat{y}_k = \sum_{j=1}^n w_{jk} \cdot F \left( \sum_{i=1}^m w_{ij} \cdot x_i + b_j \right) + c_k \quad (4)$$

where  $w_{ij}$  is the weight connecting the  $i$ -th neuron in the input layer to the  $j$ -th neuron in the hidden layer;  $b_j$  is the bias of the  $j$ -th hidden neuron;  $w_{jk}$  is the weight connecting the  $j$ -th neuron in the hidden layer to the  $k$ -th neuron in the output layer;  $c_k$  is the bias of the  $k$ -th output neuron; and  $F(\cdot)$  is the activation function of the hidden neuron. Among the various activation functions that exist, linear, sigmoid, and hyperbolic tangent functions are the most widely used functions.

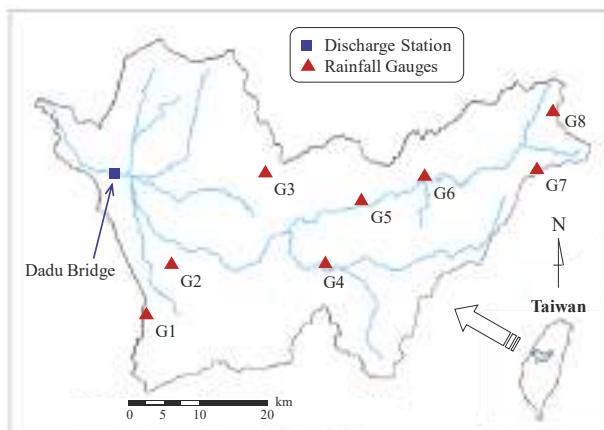
In the learning process of the back-propagation algorithm, the weights of the network are adjusted to minimize the objective function  $E$ :

$$E = \frac{1}{2} \sum_{k=1}^p (\hat{y}_k - y_k)^2 \quad (5)$$

To minimize  $E$ , the gradient descent method is used to tune the weights along the negative direction of the gradient of  $E$ . The iteration of weight adjustment is repeated until convergence is reached. The detailed process of determining the weights can be found in the literature [36,37].

## 3. Study Area and Hydrologic Data

The study area is Wu River, located in central Taiwan (Figure 4). Wu River flows through the metropolitan area of Taichung City and empties into the Taiwan Strait. Wu River encloses a basin area of 2026 km<sup>2</sup> and has a mainstream length of 119 km. The average annual precipitation in the Wu River basin is approximately 2087 mm, much of which is typhoon rainfall.



**Figure 4.** Wu River basin and locations of the gauge stations.

The downstream Dadu Bridge discharge station (Figure 4) near the metropolitan area is the forecasting object. This study collected hourly discharge data from Dadu Bridge and hourly rainfall data from eight rainfall gauges (named G1 to G8 and shown in Figure 4). Data for thirteen typhoon flood events with complete records were obtained. Among these flood events, 10 events (488 datasets) were used for calibration and three events (206 datasets) that caused flooding disasters in the downstream metropolitan area were used for validation. Table 1 lists the characteristics of the typhoon flood events, including the date and name of the typhoon, total amount of average rainfall (Thiessen polygon method) in the Dadu Bridge Basin, and peak discharge at Dadu Bridge.

**Table 1.** Characteristics of collected typhoon flood events.

Event No.	Date	Typhoon	Total Rainfall (mm)	Peak Discharge ( $\text{m}^3/\text{s}$ )	Note
01	4 August 1998	Otto	100.9	1440	Calibration
02	30 July 2001	Toraji	290.4	10,000	Calibration
03	16 September 2001	Nari	165.0	2930	Calibration
04	20 June 2012	Talim	95.5	1067	Calibration
05	1 August 2012	Saola	452.1	7199	Calibration
06	12 July 2013	Soulik	358.6	11,004	Calibration
07	21 August 2013	Trami	319.6	2011	Calibration
08	29 August 2013	Kongrey	127.0	1786	Calibration
09	7 August 2015	Soudelor	138.8	720	Calibration
10	28 September 2015	Dujuan	134.9	967	Calibration
11	31 July 1996	Herb	415.3	5630	Validation
12	1 July 2004	Mindulle	898.4	14,802	Validation
13	22 July 2014	Matmo	193.8	1704	Validation

#### 4. Model Development and Forecasting Results

##### 4.1. Determining the Input Variables

This study analyzed the lags between the discharge at Dadu Bridge with various lagged rainfall and discharge variables. The derived lagged variables were used as inputs of the proposed hybrid neural network model to forecast the discharge at Dadu Bridge. This study applied the linear transfer function (LTF) to determine the lagged variables by applying the least-squares technique to construct a linear function with lagged input variables. The *t*-test was employed to examine the statistical significance of the input variables. An advantage of using the LTF is that the lagged variables can be objectively determined by the statistical significance test. The process of using the LTF and the statistical significance test to determine the lags of input variables can be found in Chen et al. [38]. The time step for the analysis of lags and the following flood forecasting is one hour in this study.

This study used two types of hourly rainfall data as input variables: multiple rainfall data from eight rain gauges and average rainfall data. The most significant lags between the discharge at Dadu Bridge and the rainfall from each rain gauge were determined by the LTF. The lag for rain gauges G1 and G2 was 1 h, and the lag for G3 was 2 h. Rain gauges G4, G5, and G6 exhibited a lag of 3 h, and G7 and G8 exhibited a lag of 4 h. The determined lags were hydrologically rational. When the distance of a rain gauge to Dadu Bridge was longer, the most statistically significant time lag was also longer. For the average rainfall, the Thiessen polygon method was used to calculate the average rainfall in the Dadu Bridge watershed. The lagged average rainfall variables for 1–4 h were statistically significant at the 5% significance level, with the most significantly lagged variable for 3 h. The lagged discharge variables (discharge increment) were also examined. Only the lagged discharge variable for 1 h was statistically significant at the 5% significance level.

Let the hourly discharge of Dadu Bridge at the present time  $t$  be  $Q(t)$ . For flood forecasting, the forecasted discharge with the lead-time of 1 h is  $\hat{Q}(t+1)$ . When the multiple rainfall data from eight rain gauges are used as inputs according to the most significant lags, the input variables are denoted as  $R_{G1}(t), R_{G2}(t), R_{G3}(t-1), R_{G4}(t-2), R_{G5}(t-2), R_{G6}(t-2), R_{G7}(t-3)$ , and  $R_{G8}(t-3)$ , where  $R_{G1}(t)$  indicates the rainfall variable for rain gauge G1 at time  $t$ ; the same notation applies to the other rain gauges. Let the discharge increment be  $\Delta Q(t)$ , which is defined as  $\Delta Q(t) = Q(t) - Q(t-1)$ . The proposed hybrid neural network model using multiple rainfall data (denoted as  $f_I[\cdot]$ ) for forecasting the one-hour-ahead discharge increment  $\Delta \hat{Q}(t+1)$  can be formulated as

$$\Delta \hat{Q}(t+1) = f_I [\Delta Q(t), R_{G1}(t), R_{G2}(t), R_{G3}(t-1), R_{G4}(t-2), R_{G5}(t-2), R_{G6}(t-2), R_{G7}(t-3), R_{G8}(t-3)] \quad (6)$$

When the discharge increment  $\Delta \hat{Q}(t+1)$  is computed by the model and added to the observed discharge  $Q(t)$ , the one-hour-ahead discharge  $\hat{Q}(t+1)$  can be forecasted. The model that uses the basin average rainfall data  $R_A(t)$  (denoted as  $f_{II}[\cdot]$ ) is formulated as

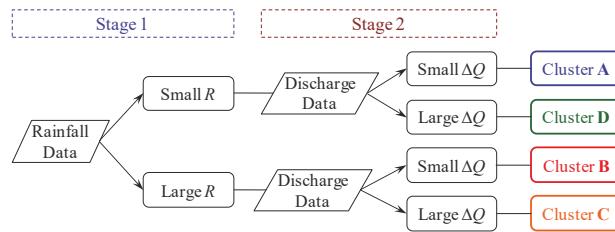
$$\Delta \hat{Q}(t+1) = f_{II} [\Delta Q(t), R_A(t), R_A(t-1), R_A(t-2), R_A(t-3)] \quad (7)$$

For convenience, the hybrid neural network model using multiple rainfall data is hereafter termed Model I, and that using average rainfall data is termed Model II.

#### 4.2. Clustering by Using the SOM

According to the proposed hybrid neural network model, input data were grouped into four hydrologically meaningful clusters formed by using the SOM. A two-stage clustering process (Figure 5) was proposed based on the properties of the rainfall and discharge data. In the first stage, input variables were grouped into two clusters (low and high rainfall clusters) by using only the rainfall data. In the second stage, the low rainfall cluster was further separated into two clusters (small and large discharge increment clusters) by using the discharge increment data. The high rainfall cluster was also divided into small and large discharge increment clusters. Consequently, four hydrologically meaningful clusters (with low and high  $R$  vs. small and large  $\Delta Q$ ) were obtained. This study applied the two-stage scheme to ensure that the rainfall and discharge increment data could be grouped into the expected four clusters. Because two forecasting models, Model I and Model II, were proposed corresponding to the two types of input rainfall variables, the clustering process was performed with respect to both the multiple rainfall data and average rainfall data.

Table 2 lists the clustering results for the calibration events (totally 488 datasets). The numbers of clusters corresponding to the two types of rainfall data are comparable. Cluster A (low  $R$  and small  $\Delta Q$ ) has the most data, and Cluster C (high  $R$  and large  $\Delta Q$ ) has the fewest data. The clustering results corresponding to the number of clusters are rational. The initial and final parts of the hydrograph (grouped as Cluster A) normally contain a large portion of the whole dataset. The rapidly rising limb of the hydrograph (grouped as Cluster C) encloses fewer data.

**Figure 5.** Process of the two-stage clustering scheme.**Table 2.** Number of clusters for the calibration data sets.

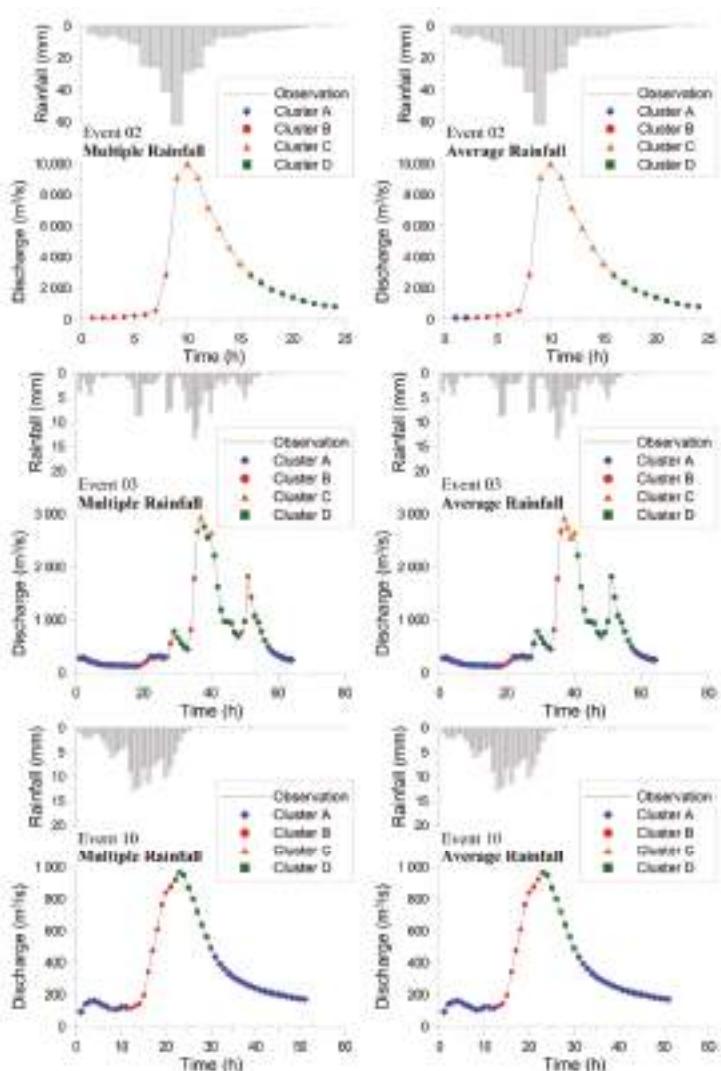
Cluster	Model I(Multiple Rainfall)	Model II(Average Rainfall)
Cluster A (Low $R$ , Small $\Delta Q$ )	222	231
Cluster B (High $R$ , Small $\Delta Q$ )	98	89
Cluster C (High $R$ , Large $\Delta Q$ )	33	35
Cluster D (Low $R$ , Large $\Delta Q$ )	135	133

Table 3 lists the ranges (minimum and maximum) of rainfall  $R$  and discharge increment  $\Delta Q$  for different clusters with respect to the average rainfall data. The average rainfall  $\leq 4.66$  mm was grouped into the low rainfall cluster, and that  $\geq 4.67$  mm was grouped into the high rainfall cluster at the first clustering stage. During the second-stage clustering process, the discharge increment data were further classified, and four clusters were obtained. The ranges of clusters are comparable to the physical meaning of clusters. According to the ranges listed in Table 3, Cluster A (low  $R$  and small  $\Delta Q$ ) has small rainfall and discharge increment data, and Cluster C (high  $R$  and large  $\Delta Q$ ) encloses the largest range among the clusters.

**Table 3.** Ranges of rainfall and discharge increment for different clusters.

Cluster	Rainfall (mm)		Discharge Increment ( $m^3/s$ )	
	Min.	Max.	Min.	Max.
Cluster A	0	4.66	-80	261
Cluster B	4.67	45.21	-303	4756
Cluster C	4.84	51.83	-3170	6280
Cluster D	0	4.56	-750	841

Figure 6 illustrates the clustering results of the SOM for the calibration events using three large, medium, and small flood events as an example. The clustering results concerning the multiple rainfall data (left panel) and average rainfall data (right panel) are similar. Event 02 (upper panel) is a large flood with a single peak caused by a concentrated and severe storm. The rainfall and discharge (also the discharge increment) data are large. Therefore, very few data (only two for the average rainfall case) are grouped as Cluster A. However, the rainfall-runoff process from Cluster B to Clusters C and D is appropriately identified by the clustered data. Event 03 (middle panel) is a medium flood with multiple peaks caused by a series of intermittent storms. The hydrologic process shown by the clusters is somewhat complicated. Nevertheless, progress from Cluster A to Clusters B, C, and D can be observed, with some data of Cluster C in the peak segment. Event 10 (lower panel) is an event with small peak discharge and low rainfall. The hydrograph is favorably described by the clusters; however, no data is grouped as Cluster C due to the low rainfall and discharge.



**Figure 6.** Clustering results of the self-organizing map (SOM) for three calibration events.

The SOM clustering was validated using the validation events, and the results are as follows. Two of the three events that caused flooding disasters in the metropolitan area were relatively large flood events. The greater numbers of Clusters C and D (with large discharge increments) than those of Clusters A and B (with small discharge increments) listed in Table 4 indicate the circumstances. Figure 7 presents the clustering results for the validation events. Event 11 is a large flood, with lots of data around the peak grouped as Cluster C and no data grouped as Cluster A. Event 12 is an extremely large flood with a peak discharge much higher than is present in the calibration data (cf. Table 1). Many high discharge data were reasonably classified as Cluster C. The clustering results show a clear progression from Cluster A to Clusters B, C, and D for the first hydrograph, and also an obvious sequence of Clusters B, C, D, and A for the second hydrograph. Event 13 is a small flood event that shows a similar result as Event 10 in the calibration set. The hydrograph is well explained by the

clusters; however, no data is grouped as Cluster C. The calibration and validation results prove that the proposed clustering method based on the hydrologic process meaningfully depicts the physical processes behind rainfall and discharge data.

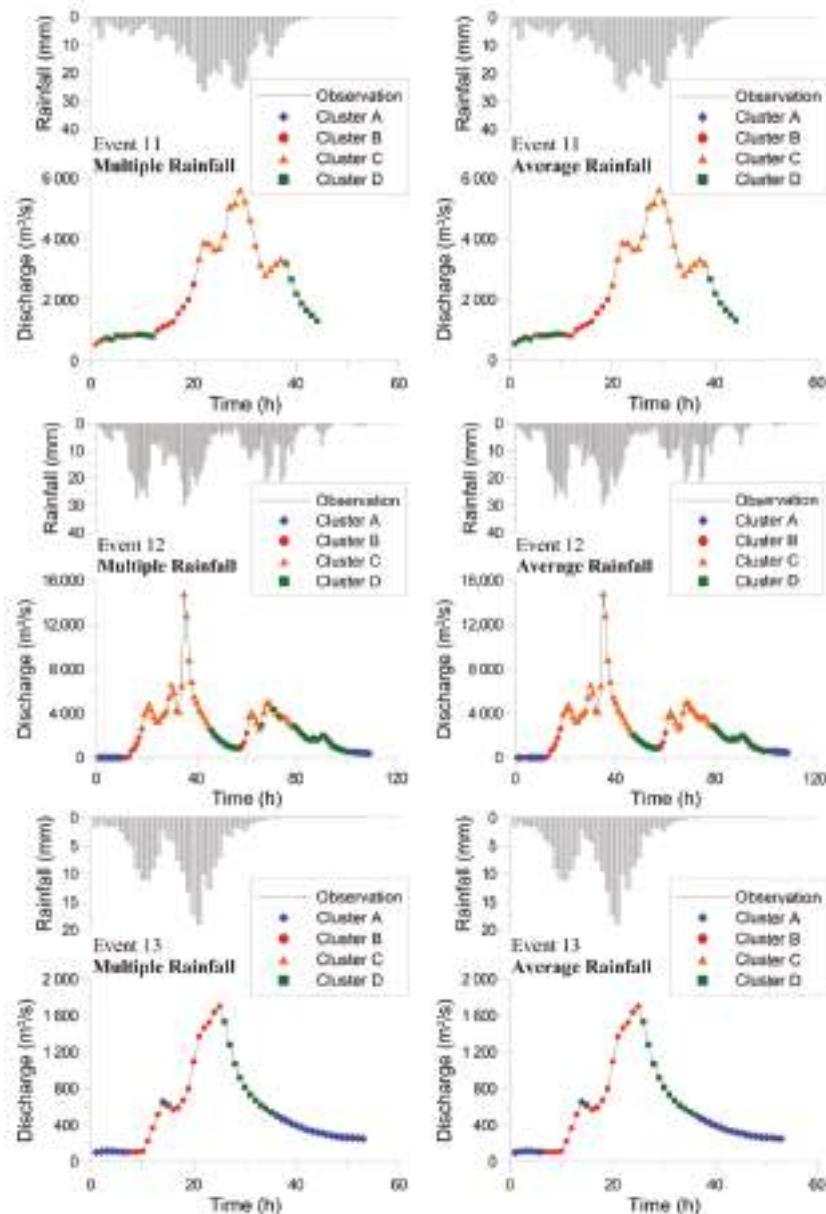


Figure 7. Clustering results of the SOM for the validation events.

**Table 4.** Number of clusters for the validation datasets.

Cluster	Model I (Multiple Rainfall)	Model II (Average Rainfall)
Cluster A (Low $R$ , Small $\Delta Q$ )	42	43
Cluster B (High $R$ , Small $\Delta Q$ )	40	40
Cluster C (High $R$ , Large $\Delta Q$ )	59	60
Cluster D (Low $R$ , Large $\Delta Q$ )	65	62

#### 4.3. Flood Forecasting Using the Hybrid Neural Network Model

For each cluster, BPNNs were constructed with respect to the structures of Model I and Model II. The calibration data used in constructing the BPNNs were linearly normalized to the interval between 0 and 1 according to the minimum and maximum values in the calibration data. The hidden nodes and activation functions of the BPNNs were determined through trial and error. Table 5 lists the calibration results regarding the number of hidden nodes and types of activation functions. The numbers of hidden nodes of the BPNNs using multiple rainfall data are generally greater than those of the BPNNs using average rainfall data. The multiple rainfall data possess a more complex spatial pattern than the average rainfall data. Therefore, more hidden nodes are required to describe the complex relationship between the inputs and output. An interesting result is the derived activation functions. The BPNNs corresponding to Cluster A (small  $R$  and small  $\Delta Q$ ) and Cluster C (large  $R$  and large  $\Delta Q$ ) use the linear function. The BPNNs for Cluster B (large  $R$  and small  $\Delta Q$ ) and Cluster D (small  $R$  and large  $\Delta Q$ ) use the sigmoid function. When the rainfall and discharge increment data in a cluster have similar properties (i.e., all small or all large), the linear function is sufficient to model the input–output relationship. When the data in a cluster are different (i.e., small vs. large), the nonlinear sigmoid function is used to model the complex input–output relationship.

**Table 5.** Calibrated numbers of hidden nodes and types of activation functions.

Cluster	Model I		Model II	
	Number of Hidden Nodes	Activation Function	Number of Hidden Nodes	Activation Function
Cluster A (Low $R$ , Small $\Delta Q$ )	3	Linear	2	Linear
Cluster B (High $R$ , Small $\Delta Q$ )	3	Sigmoid	2	Sigmoid
Cluster C (High $R$ , Large $\Delta Q$ )	2	Linear	2	Linear
Cluster D (Low $R$ , Large $\Delta Q$ )	4	Sigmoid	2	Sigmoid

With the constructed SOMs and BPNNs, the hybrid neural network model was established for flood forecasting with respect to the calibration and validation events. Performance indices—the coefficient of efficiency (CE), mean absolute error (MAE), and error of time to peak discharge (ETP)—were obtained as follows:

$$CE = 1 - \frac{\sum_{t=1}^n (Q(t) - \hat{Q}(t))^2}{\sum_{t=1}^n (Q(t) - \bar{Q})^2} \quad (8)$$

$$MAE = \frac{\sum_{t=1}^n |Q(t) - \hat{Q}(t)|}{n} \quad (9)$$

$$ETP = \hat{T}_p - T_p \quad (10)$$

where  $Q(t)$  is the observed discharge at time  $t$ ;  $\hat{Q}(t)$  is the forecasted discharge;  $\bar{Q}$  is the average observed discharge;  $n$  is the number of data;  $T_p$  is the time to peak for observed discharge; and  $\hat{T}_p$  is

the time to peak for forecasted discharge. CE is a dimensionless index with a value of unity indicating perfect fit. MAE is an index that directly describes the average forecast error with the same unit of the data. ETP is positive if the forecasted peak discharge is delayed. This lag often exists in hydrological forecasting. The model that has a smaller absolute value of ETP is better in forecasting performance.

Table 6 lists the performance indices of the hybrid neural network model for the calibration and validation events. The CE values for the calibration data are 0.97 and 0.98 corresponding to Model I and Model II, respectively, whereas the MAE values are 92.9 and 68.2 m<sup>3</sup>/s, which are small compared to the discharge magnitude of the flood events. For the validation events, CE is 0.94 and 0.91 and MAE is 188.0 and 248.2 m<sup>3</sup>/s, respectively. ETPs for calibration events range from −2 to 1 h; ETP is zero for half of the events. The average ETPs are small as shown in Table 6. The performance indices prove that the proposed hybrid neural network model favorably forecasts the flood discharge and that the performance of Model I and Model II is similar.

**Table 6.** Performance indices of the hybrid neural network model.

Data	Model Type	CE	MAE (m <sup>3</sup> /s)	ETP (h)
Calibration	Model I	0.97	92.9	−0.2
	Model II	0.98	68.2	−0.1
Validation	Model I	0.94	188.0	0.3
	Model II	0.91	248.2	0.0

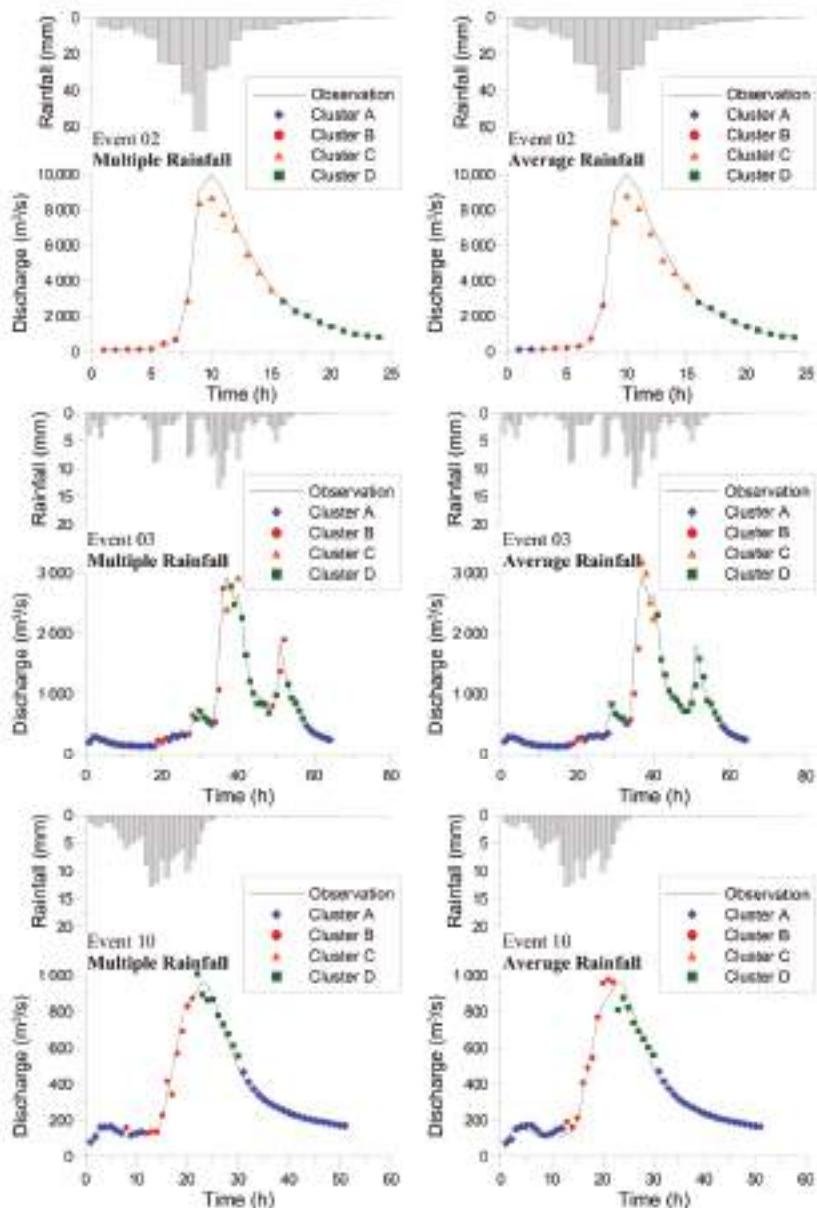
Figures 8 and 9 present the forecasted hydrographs for the calibration and validation events, respectively. In general, the forecasted hydrograph matches the observed hydrograph. However, the forecasted discharge for Cluster C is not as close as that for the other clusters. During the model learning process, only 33 and 35 datasets were used to train the BPNNs for Cluster C (Table 2). Although the BPNNs trained using fewer data have larger errors, Event 12 has a peak discharge much higher than the calibration data. The forecasted discharges around the crest segment are reasonable, indicating that the hybrid neural network model extrapolates successfully. Overall, the forecasting results demonstrate that the proposed hybrid neural network model accurately forecasts typhoon floods, including small, medium, and large events, and the two types of model (Model I and Model II) have comparable capability.

#### 4.4. Comparison with Traditional Neural Network Model

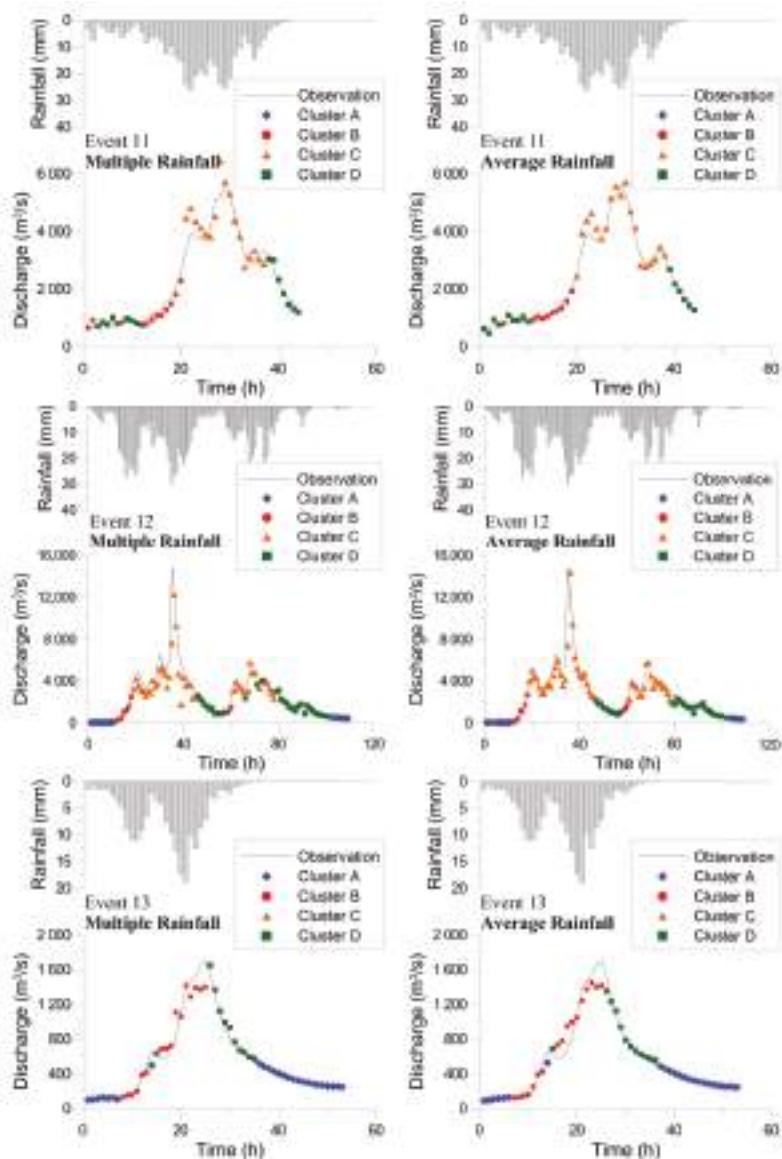
This study also developed a traditional neural network model to assess and compare its performance with that of the hybrid neural network model. The traditional neural network model, which does not group data into clusters, uses all the calibration data to construct a single BPNN using the same calibration scheme as the hybrid neural network model. The single BPNN was also trained by using the two types of rainfall variable (Model I and Model II). The constructed traditional BPNNs have three hidden nodes that use the sigmoid activation function. Table 7 lists the performance indices of the traditional neural network model. The CE value for calibration is 0.95, which is a little lower than the CE values (0.97 and 0.98) of the hybrid neural network model. However, the CE value of 0.85 for validation is considerably lower than those (0.94 and 0.91) of the hybrid neural network model. The MAE and the ETP values of the traditional BPNN are larger than those of the hybrid neural network model (cf. Tables 6 and 7). Although the traditional BPNN also exhibits good forecasting performance in view of the performance indices, the hybrid neural network model apparently outperform the traditional neural network model.

Figure 10 displays the forecasted hydrographs obtained using the hybrid and traditional neural network models pertaining to the validation events. In general, the two sets of the forecasted hydrographs have similar patterns. However, the hydrographs obtained using the traditional BPNN exhibit minor underestimation for large discharges (especially for the peak discharge in Event 12) and

overestimation for small discharges (especially for the case in Event 13). The traditional BPNN was trained using small and large discharge data simultaneously. The learning mechanism matches the whole calibration data. Thus, the single BPNN does not perform very well in some cases of small and large discharges. However, the proposed hybrid neural network model was trained using different clusters with respective small and large datasets. The hybrid neural network model is more robust and flexible for various rainfall-runoff processes.



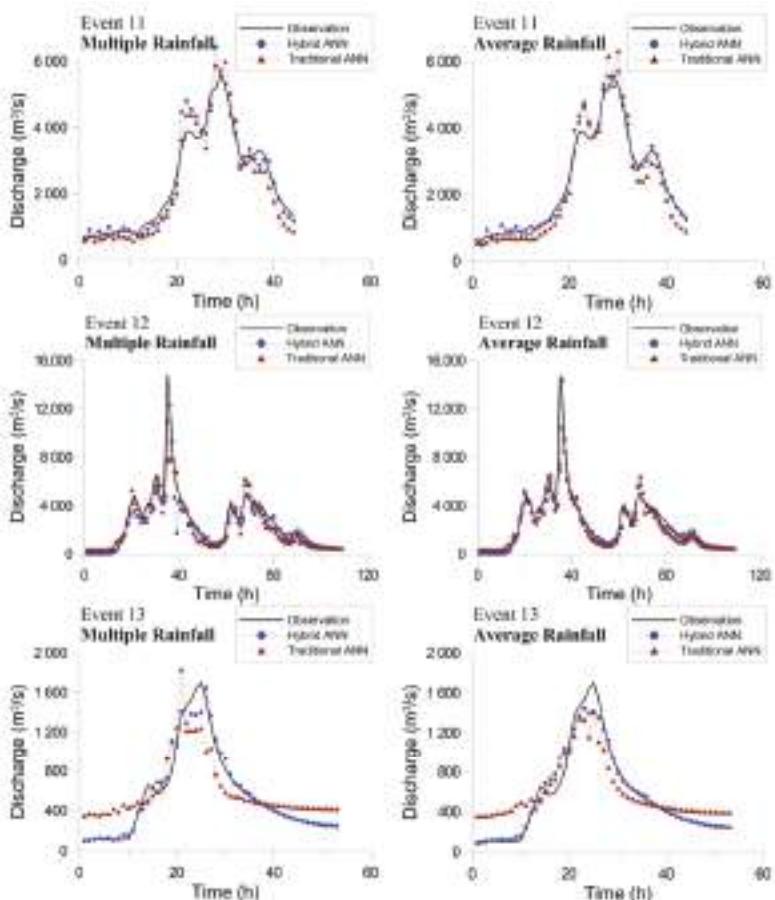
**Figure 8.** Flood forecasting results of the proposed model for calibration events.



**Figure 9.** Flood forecasting results of the proposed model for validation events.

**Table 7.** Performance indices of the traditional neural network model.

Data	Model Type	CE	MAE ( $\text{m}^3/\text{s}$ )	ETP (h)
Calibration	Model I	0.95	229.2	-0.1
	Model II	0.95	242.4	-0.5
Validation	Model I	0.85	339.8	-1.0
	Model II	0.85	359.1	0.7



**Figure 10.** Comparison of the flood forecasting results for the hybrid and traditional neural network models.

## 5. Conclusions

ANNs, usually regarded as black boxes, suffer from a lack of physical interpretation of the constructed model architecture. This study proposed a physical hybrid neural network model that combines the SOM and BPNNs and applied this proposed model to real-time flood forecasting. The SOM was used to group the rainfall and discharge data into four clusters with clear physical meanings to characterize the rainfall-runoff process. Then, a BPNN was constructed for each cluster with specific properties of rainfall and discharge data, which gave the BPNNs higher capability and a network structure that could be meaningfully discussed.

Typhoon flood discharges at Dadu Bridge and rainfall from eight rain gauges in the Wu River basin in Taiwan were used as the study data. Two types of rainfall data (multiple rainfall and average rainfall) were used to construct two types of hybrid neural network model (Model I and Model II). The lagged input variables of the models were determined by using the LTF. The derived lags of the rainfall variables are hydrologically rational and represent the distance from the rain gauge to the basin outlet.

The clustering results of the SOM pertaining to calibration and validation events prove that the hydrologic process is meaningfully described by the clusters. The rainfall-runoff process can

be identified by the sequence of Clusters A, B, C, D, and A. The training of the BPNNs reveals that more hidden nodes are required to describe the complex relationship between multiple rainfall and discharge. The simple linear activation function was adopted in the clusters with similar data, whereas the nonlinear sigmoid activation function was used in clusters where the rainfall and discharge data were different.

Flood forecasting using the hybrid neural network model revealed that the proposed model successfully forecasts flood discharge with high efficiency and small errors. Both Model I and Model II have comparable forecasting performance. This study also developed a traditional neural network for comparison with the hybrid neural network model. The traditional neural network model that was trained with the whole calibration dataset did not perform favorably in some cases of small and large discharges. With respect to the performance indices and forecast hydrographs, the proposed physical hybrid neural network model exhibits robust flood forecasting and outperforms the traditional neural network model.

**Author Contributions:** Y.-D. Jhong and C.-S. Chen designed the research; H.-P. Lin and Y.-D. Jhong conducted the analysis; C.-S. Chen and S.-T. Chen supervised the study; Y.-D. Jhong and S.-T. Chen wrote the manuscript. All authors read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Chang, F.J.; Chiang, Y.M.; Chang, L.C. Multi-step-ahead neural networks for flood forecasting. *Hydrol. Sci. J.* **2007**, *52*, 114–130. [[CrossRef](#)]
- Dawson, C.W.; Wilby, R. An artificial neural network approach to rainfall-runoff modelling. *Hydrol. Sci. J.* **1998**, *43*, 47–66. [[CrossRef](#)]
- French, M.N.; Krajewski, W.F.; Cuykendall, R.R. Rainfall forecasting in space and time using a neural network. *J. Hydrol.* **1992**, *137*, 1–31. [[CrossRef](#)]
- Lin, G.F.; Wu, M.C. A hybrid neural network model for typhoon-rainfall forecasting. *J. Hydrol.* **2009**, *375*, 450–458. [[CrossRef](#)]
- Lin, G.F.; Jhong, B.C.; Chang, C.C. Development of an effective data-driven model for hourly typhoon rainfall forecasting. *J. Hydrol.* **2013**, *495*, 52–63. [[CrossRef](#)]
- See, L.; Openshaw, S. Applying soft computing approaches to river level forecasting. *Hydrol. Sci. J.* **1999**, *44*, 763–778. [[CrossRef](#)]
- Young, C.C.; Liu, W.C. Prediction and modelling of rainfall-runoff during typhoon events using a physically-based and artificial neural network hybrid model. *Hydrol. Sci. J.* **2015**, *60*, 2102–2116. [[CrossRef](#)]
- Chen, S.T.; Yu, P.S.; Liu, B.W. Comparison of neural network architectures and inputs for radar rainfall adjustment for typhoon events. *J. Hydrol.* **2011**, *405*, 150–160. [[CrossRef](#)]
- Bray, M.; Han, D. Identification of support vector machines for runoff modeling. *J. Hydroinf.* **2004**, *6*, 265–280.
- Chen, S.T. Multiclass support vector classification to estimate typhoon rainfall distribution. *Disaster Adv.* **2013**, *6*, 110–121.
- Chen, S.T.; Yu, P.S. Real-time probabilistic forecasting of flood stages. *J. Hydrol.* **2007**, *340*, 63–77. [[CrossRef](#)]
- Chen, S.T.; Yu, P.S. Pruning of support vector networks on flood forecasting. *J. Hydrol.* **2007**, *347*, 67–78. [[CrossRef](#)]
- Han, D.; Chan, L.; Zhu, N. Flood forecasting using support vector machines. *J. Hydroinf.* **2007**, *9*, 267–276. [[CrossRef](#)]
- Lin, G.F.; Chen, G.R.; Huang, P.Y.; Chou, Y.C. Support vector machine-based models for hourly reservoir inflow forecasting during typhoon-warning periods. *J. Hydrol.* **2009**, *372*, 17–29. [[CrossRef](#)]
- Liong, S.Y.; Sivapragasam, C. Flood stage forecasting with support vector machines. *J. Am. Water Resour. Assoc.* **2002**, *38*, 173–186. [[CrossRef](#)]
- Yu, P.S.; Chen, S.T.; Chang, I.F. Support vector regression for real-time flood stage forecasting. *J. Hydrol.* **2006**, *328*, 704–716. [[CrossRef](#)]
- Yu, P.S.; Yang, T.C.; Chen, S.Y.; Kuo, C.M.; Tseng, H.W. Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting. *J. Hydrol.* **2017**, *552*, 92–104. [[CrossRef](#)]

18. Ashrafi, M.; Chua, L.H.C.; Quek, C.; Qin, X. A fully-online Neuro-Fuzzy model for flow forecasting in basins with limited data. *J. Hydrol.* **2017**, *545*, 424–435. [[CrossRef](#)]
19. Chang, F.J.; Chen, Y.C. A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction. *J. Hydrol.* **2001**, *245*, 153–164. [[CrossRef](#)]
20. Lohani, A.K.; Kumar, R.; Singh, R.D. Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. *J. Hydrol.* **2012**, *442*, 23–35. [[CrossRef](#)]
21. Mukerji, A.; Chatterjee, C.; Raghuwanshi, N.S. Flood forecasting using ANN, neuro-fuzzy, and neuro-GA models. *J. Hydrol. Eng.* **2009**, *14*, 647–652. [[CrossRef](#)]
22. Nayak, P.C.; Sudheer, K.P.; Rangan, D.M.; Ramasastri, K.S. A neuro-fuzzy computing technique for modeling hydrological time series. *J. Hydrol.* **2004**, *291*, 52–66. [[CrossRef](#)]
23. Nayak, P.C.; Sudheer, K.P.; Rangan, D.P.; Ramasastri, K.S. Short-term flood forecasting with a neurofuzzy model. *Water Resour. Res.* **2005**, *41*, W04004. [[CrossRef](#)]
24. Nayak, P.C.; Sudheer, K.P.; Jain, S.K. Rainfall-runoff modeling through hybrid intelligent system. *Water Resour. Res.* **2007**, *43*, W07415. [[CrossRef](#)]
25. Yarar, A. A hybrid wavelet and neuro-fuzzy model for forecasting the monthly streamflow data. *Water Resour. Manag.* **2014**, *28*, 553–565. [[CrossRef](#)]
26. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [[CrossRef](#)]
27. Lange, N.T. New mathematical approaches in hydrological modeling—an application of artificial neural networks. *Phys. Chem. Earth Part B* **1999**, *24*, 31–35. [[CrossRef](#)]
28. Jain, A.; Sudheer, K.P.; Srinivasulu, S. Identification of physical processes inherent in artificial neural network rainfall runoff models. *Hydrol. Process.* **2004**, *18*, 571–581. [[CrossRef](#)]
29. Chen, S.T. Mining informative hydrologic data by using support vector machines and elucidating mined data according to information entropy. *Entropy* **2015**, *17*, 1023–1041. [[CrossRef](#)]
30. Furundzic, D. Application example of neural networks for time series analysis: Rainfall–runoff modeling. *Signal Process.* **1998**, *64*, 383–396. [[CrossRef](#)]
31. Abrahart, R.J.; See, L. Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrol. Process.* **2000**, *14*, 2157–2172. [[CrossRef](#)]
32. Hsu, K.L.; Gupta, H.V.; Gao, X.; Sorooshian, S.; Imam, B. Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. *Water Resour. Res.* **2002**, *38*, 38–1–38–17. [[CrossRef](#)]
33. Jain, A.; Srinivasulu, S. Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. *J. Hydrol.* **2006**, *317*, 291–306. [[CrossRef](#)]
34. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 59–69. [[CrossRef](#)]
35. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
36. Hagan, M.T.; Demuth, H.B.; Beale, M.H. *Neural Network Design*; PWS Publishing: Boston, MA, USA, 1996.
37. Haykin, S. *Neural Networks: A Comprehensive Foundation*; MacMillan: New York, NY, USA, 1994.
38. Chen, C.S.; Chen, B.P.T.; Chou, F.N.F.; Yang, C.C. Development and application of a decision group back-propagation neural network for flood forecasting. *J. Hydrol.* **2010**, *385*, 173–182. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

## Article

# Improving the Muskingum Flood Routing Method Using a Hybrid of Particle Swarm Optimization and Bat Algorithm

Mohammad Ehteram <sup>1</sup>, Faridah Binti Othman <sup>2</sup>, Zaher Mundher Yaseen <sup>3,\*</sup>,  
Haitham Abdulmohsin Afan <sup>3</sup>, Mohammed Falah Allawi <sup>4</sup>, Marlinda Bt. Abdul Malek <sup>5,6</sup>,  
Ali Najah Ahmed <sup>5,7</sup>, Shamsuddin Shahid <sup>8</sup>, Vijay P. Singh <sup>9</sup> and Ahmed El-Shafie <sup>2</sup>

<sup>1</sup> Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan 35131-19111, Iran; mohammdehteram@semnan.ac.ir

<sup>2</sup> Department of Civil Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia; faridahothman@um.edu.my (F.B.O); elshafie@um.edu.my (A.E.-S.)

<sup>3</sup> Sustainable Developments in Civil Engineering Research Group, Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam; haitham.afan@gmail.com

<sup>4</sup> Civil and Structural Engineering Department, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia; mohmmd.falah@gmail.com

<sup>5</sup> Department of Civil Engineering, College of Engineering, University Tenaga National, Kajang 43000, Malaysia; Marlinda@uniten.edu.my (M.B.A.M.); Mahfoodh@uniten.edu.my (A.N.A.);

<sup>6</sup> Institute of Policy Energy Research (IPERe), Universiti Tenaga National, Kajang 43000, Malaysia

<sup>7</sup> Institute for Energy Infrastructures, University Tenaga National, Kajang 43000, Malaysia

<sup>8</sup> Faculty of Civil Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia; sshahid@utm.my

<sup>9</sup> Department of Biological and Agricultural Engineering and Zachry Department of Civil Engineering, Texas A&M University, 2117 TAMU, College Station, TX 77843-2117, USA; vsingh@tamu.edu

\* Correspondence: yaseen@tdt.edu.vn; Tel.: +84-163-498-7030

Received: 21 May 2018; Accepted: 15 June 2018; Published: 19 June 2018

**Abstract:** Flood prediction and control are among the major tools for decision makers and water resources planners to avoid flood disasters. The Muskingum model is one of the most widely used methods for flood routing prediction. The Muskingum model contains four parameters that must be determined for accurate flood routing. In this context, an optimization process that self-searches for the optimal values of these four parameters might improve the traditional Muskingum model. In this study, a hybrid of the bat algorithm (BA) and the particle swarm optimization (PSO) algorithm, i.e., the hybrid bat-swarm algorithm (HBSA), was developed for the optimal determination of these four parameters. Data for the three different case studies from the USA and the UK were utilized to examine the suitability of the proposed HBSA for flood routing. Comparative analyses based on the sum of squared deviations (SSD), sum of absolute deviations (SAD), error of peak discharge, and error of time to peak showed that the proposed HBSA based on the Muskingum model achieved excellent flood routing accuracy compared to that of other methods while requiring less computational time.

**Keywords:** bat algorithm; particle swarm optimization; flood routing; Muskingum model

## 1. Introduction

Floods cause huge economic and social effects on the surrounding environment [1,2], such as breaking levees [3], inundating houses, disrupting transportation systems [4], damaging crops and eroding fertile lands [5]. Thus, flood prediction and flood control are important issues for policy makers and designers [6,7]. Hydrological and hydraulic models are used for flood prediction. The estimation of flood discharge hydrograph at a downstream location given the discharge hydrograph upstream is

known as flood routing [8,9]. Hydraulic models based on numerical methods can be used for flood routing, but they involve complex unsteady flow equations [10]. Hydrological models use the spatially lumped continuity equation and a storage equation for flood routing. These models need a small amount of data to predict floods [11]. The Muskingum model is an important hydrological model for flood routing. This model has multiple parameters that should be obtained to accurately predict floods [12], and different versions of the model have been applied for flood routing. One strategy is the optimization method by which the parameters are computed as decision variables [13]. The evolutionary algorithms exhibit a high degree of ability for solving complex optimization problems. These algorithms are efficient, accurate, and flexible [14]. Thus, parameters of the Muskingum model can be computed using evolutionary algorithms.

### 1.1. Background

Luo and Xie [15] applied the clonal selection algorithm (CLA) to flood routing in China, and CLA decreased the sum of absolute deviations (SAD) by approximately 20% compared to that of the genetic algorithm (GA). The difference between the simulated peak discharge and the observed discharge was small, approximately  $0.5 \text{ m}^3/\text{s}$ . Using the initial values of the Muskingum parameters as decision variables, CLA optimized the objective function to obtain the best parameter values.

The Nelder-Mead simplex algorithm was used for flood routing in the USA [16]. This method obtained the best values for the 3 Muskingum model parameters. Thus, the peak discharges were predicted to be better than those of other nonlinear programming methods.

Barati et al. [17] applied a dynamic wave method for flood routing, and the results indicated that some of the parameters, such as Manning roughness or bed slope, were effective. Thus, accurate determination of these parameters is important for flood routing equations.

Karahan et al. [18] applied hybrid harmony search (HS) and particle swarm optimization (PSO) to estimate the Muskingum parameters and indicated that the computational time for the new hybrid model was less than those of PSO and HS. In addition, peak discharge was estimated with the least difference with the observed discharge.

Fallah-Mehdipour [19] applied PSO, GA and nonlinear programming methods for flood routing and indicated that PSO simulated the flood discharges with lower SAD and sum of squared deviation (SSD) values than those of other algorithms. In PSO, the Muskingum parameters were considered as decision variables.

Easa [20] introduced a four-parameter nonlinear Muskingum model for flood routing and showed that the four-parameter model estimated flood discharge better than the three-parameter model did. The GA was used to obtain the optimal values of the Muskingum parameters.

Nelder-simplex and hybrid PSO were evaluated as new methods to estimate Muskingum parameters [21]. The results showed that the new method decreased the computational time compared to that of the simple PSO and GA. Additionally, there was a small difference between the estimated discharge and the observed discharge.

Honey bee mating (HBM) optimization was used for flood routing by a three-parameter nonlinear Muskingum model for one flood in the USA [22]. Results indicated that the SSQ (sum of squared deviations) and SAD values of the model significantly decreased compared to those of the GA and PSO methods. Additionally, the time of peak discharge was predicted well. The convergence of HBM was faster than those of GA and PSO. A new charged search system (CSS) and PSO hybrid method was used for flood routing based on two- and three-parameter nonlinear Muskingum models [23]. Results indicated that the new evolutionary hybrid algorithm needed a sensitivity analysis for the accurate determination of parameters, and the three-parameter model predicted peak discharge more accurately than GA and HS. Additionally, basing PSO and CSS on more population diversity increased the convergence speed of the algorithm.

A nonlinear Muskingum model that considers lateral flow was used for flood routing [24]. This model used the cuckoo algorithm and predicted peak discharge more accurately than other

models. The shuffled frog leaping algorithm (SFLA) based on a three-parameter Muskingum model was used for flood routing [25]. Results showed that the SSD and SAD values for the SFLA were lower than those of the GA, PSO and nonlinear programming methods. Additionally, the correlation coefficient based on the SFA between the observed and estimated discharges was more than those of the GA, PSO and nonlinear programming methods.

A new version of the Muskingum model with nine parameters was used for flood routing in another study [26]. The Karun River in Iran was considered as an important case study for the research. The study showed that the results were better than those of previous Muskingum models, but the new model needed more computational time.

A real-coded adaptive GA was used for flood routing, considering lateral flow along the river reach [14]. The study considered a river in China with important floods. Results indicated that the three-parameter nonlinear Muskingum model based on adaptive GA simulated flood discharges, yielded the least values of SSQ and SAD.

Gene-expression programming (GEP) and the weed algorithm (WA) were used for flood routing [1]. Some important floods in the USA were considered as case studies for the study. The results indicated that GEP had a convergence speed that was approximately 100 times higher than that of WA; in addition, the computed SSD for the estimated discharges was lower than that of WA.

The literature review shows that the evolutionary algorithms have high ability for predicting flood routing, but there are some limitations for these algorithms; they all use the same procedure within the Muskingum model for flood routing. The Muskingum models have parameters with unknown values, and these parameters are applied to the algorithms as decision variables. Computation of the objective function can show the best value for each parameter. In fact, the parameters are input as the initial population to the algorithms, and then, the different operators are applied based on the process of each algorithm [1,10,22]. Basically, there are two major limitations: First, there is no assurance that the computed value for Muskingum's parameters are optimally achieved, second, the convergence rate was relatively high as the required computed values are four parameters.

For example, some algorithms such as the GA can become trapped in the local optimums and cannot compute the best values for the parameters [10]. Some literature reviews have reported that some algorithms have immature solutions because the convergence process happens abruptly as with PSO [21]. Some algorithms (e.g., GA [10]) require more computational time for the convergence process, and some algorithms such as Anti Bee Colony and Shark Algorithm need to accurately determine many random parameters, which leads to complex processes for optimization. Thus, the presentation of a better method that could hybridize the advantages of two different methods and hence has the potential to overcome those drawbacks and achieve better results with high convergence rate is necessary. The present study attempts to develop one of the known algorithms based on a hybrid process for the flood routing. The Bat Algorithm (BA) is developed based on the motivation that the algorithm has problems such as trapping in the local optimums and slow convergence, and the algorithm is based on the high ability of bats to perceive sounds [21]. Thus, the PSO is suggested for the hybrid process to improve the BA, and the hybrid process prevents the abrupt convergence of the PSO. The next section presents the innovation and objectives of this study. Then, the optimization methods are explained, and the results and conclusions for three case studies are explained in the following sections.

## 1.2. Innovation and Objectives

The bat algorithm (BA), as an optimization method, is based on living bats and the powerful ability of bats to receive sounds from their surroundings. It is used widely in different fields of image processing [27], data sensing systems [28], the determination of the seismic safety of structures [29], the design of wireless sensors [30], and water resource management [31]. However, the algorithm has some weaknesses, such as the probability of being trapped in the local optimums and slow convergence in some complex engineering problems, so it is necessary to modify the BA. This paper reports on a

hybrid algorithm (HA) based on PSO and BA. The new HA substitutes the weaker BA solution with the best PSO solution, which can prevent trapping in the local optimums and increase the convergence speed. Additionally, with the hybrid model, the BA population diversity and exploration ability increase as the optimal solution is obtained. These improvements are the motivations underlying the introduction of the new HA in this paper. The new HA was used for flood routing by a four-parameter Muskingum model in two case studies. These two case studies have been investigated widely as benchmarks in the literature, so there is comprehensive information for the comparison of the new HA with other evolutionary algorithms. This study used the HA to find the optimal values of the four parameters of the Muskingum model, extracted the output hydrographs for two case studies, and then compared the results with those generated by other types of Muskingum models and other evolutionary algorithms used in previous studies.

## 2. Methods

### 2.1. Muskingum Model

The linear Muskingum model is based on the continuity equation and a storage equation [32–34]:

$$\frac{ds_t}{dt} = I_t - O_t \quad (1)$$

$$S_t = K[xI_t + (1-x)O_t] \quad (2)$$

where  $S_t$  is the storage ( $L^3$ );  $I_t$  is the inflow ( $L^3 \cdot T^{-1}$ );  $O_t$  is the outflow ( $L^3 \cdot T^{-1}$ );  $K$  is the storage time constant, which varies from 0 to 30 (T); and  $x$  is the weighting factor, which varies from 0 to 0.5. Previous studies have shown that the linear Muskingum model does not perform well for some rivers; thus, nonlinear Muskingum models have been suggested [33]:

$$S_t = K_t[xI_t + (1-x)O_t]^m \quad (3)$$

$$S_t = K[xI_t^m + (1-x)O_t^m] \quad (4)$$

The models of Equations (3) and (4) have an additional parameter ( $m$ ), and the dimension of  $K$  is ( $L^{3(1-m)} \cdot T^m$ ). The current study uses a four-parameter nonlinear Muskingum model based on the model introduced by Easa [20], with a reported high flood routing ability:

$$S_t = K[xI_t^\alpha + (1-x)O_t^\alpha]^m \quad (5)$$

Equation (5) is similar to Equation (3). However, in Equation (5), parameter  $m$  is related to the nonlinear form of the storage equation, while in Equation (3),  $m$  is related to the linear form of the storage equation. Equation (5) can cover previous equations; that is, if  $\alpha = m = 1$ , Equation (5) will be the same as Equation (2). If  $m = 1$  (or)  $\alpha = 1$ , Equation (5) covers Equations (3) and (4). The outflow for Equation (3) is computed as follows:

$$O_t = \left[ \left( \frac{1}{1-x} \right) \left( \frac{S_t}{K} \right)^{\frac{1}{m}} - \left( \frac{x}{1-x} \right) \right]^{\frac{1}{\alpha}} \quad (6)$$

Then, Equation (6) is inserted into Equation (1), and the change in storage is computed based on the following equation:

$$\frac{\Delta S_t}{\Delta t} = I_t - \left[ \left( \frac{1}{1-x} \right) \left( \frac{S_t}{K} \right)^{\frac{1}{m}} - \left( \frac{x}{1-x} \right) I_t^\alpha \right]^{\frac{1}{\alpha}} \quad (7)$$

Finally, the storage for the next step time is computed based on the following equation:

$$S_{t+1} = S_t + \Delta t \quad (8)$$

## 2.2. Bat Algorithm

The BA is based on the powerful echolocation ability of bats, which can generate loud sounds and receive the echoes of the sounds as they return from the surroundings. The BA is based on the following assumptions [31]:

- (1) All bats use echolocation to identify prey and obstacles based on received sound frequencies.
- (2) All bats fly randomly with the velocity ( $v_l$ ) at position ( $y_l$ ), and the frequency, loudness and wavelength values are  $f_l$ ,  $A_0$  and  $\lambda$ , respectively.
- (3) The loudness changes from a large positive ( $A_0$ ) to a small positive value ( $A_{\min}$ ).

The sounds generated by bats have a pulsation rate ( $r_l$ ) that varies from 0 to 1. The value 1 means that the pulsation rate has reached a maximum value, and 0 means that the pulsation rate has reached a minimum value. The velocity, frequency and position are updated based on the following equations [31]:

$$f_l = f_{\min} + (f_{\max} - f_{\min}) \times \beta \quad (9)$$

$$v_l(t) = [y_l(t-1) - Y_*] \times f_l \quad (10)$$

$$y_l(t) = y_l(t-1) + v_l(t) \times t \quad (11)$$

where  $f_l$  is the frequency,  $f_{\min}$  is the minimum frequency,  $f_{\max}$  is the maximum frequency,  $Y_*$  is the best position for the bats,  $t$  is the time step,  $y_l(t-1)$  is the position of the bats at time  $t-1$ ,  $v_l(t)$  is the velocity, and  $\beta$  is the random vector.

A random walk is considered as the local search for the BA:

$$y(t) = y(t-1) + \varepsilon A(t) \quad (12)$$

where  $\varepsilon$  is the random value between -1 and 1; and  $A(t)$  is the loudness.

The pulsation rate and loudness are updated for each level. When the bats find prey, the pulsation rate increases and the loudness decreases for each level. The pulsation rate is updated, based on the following equation:

$$r_l^{t+1} = r_l^0 [1 - \exp(-\gamma t)] A_l^{t+1} = \alpha A_l^t \quad (13)$$

where  $\alpha$  and  $\gamma$  are the constant parameters. The BA is shown in Figure 1.

## 2.3. Particle Swarm Optimization

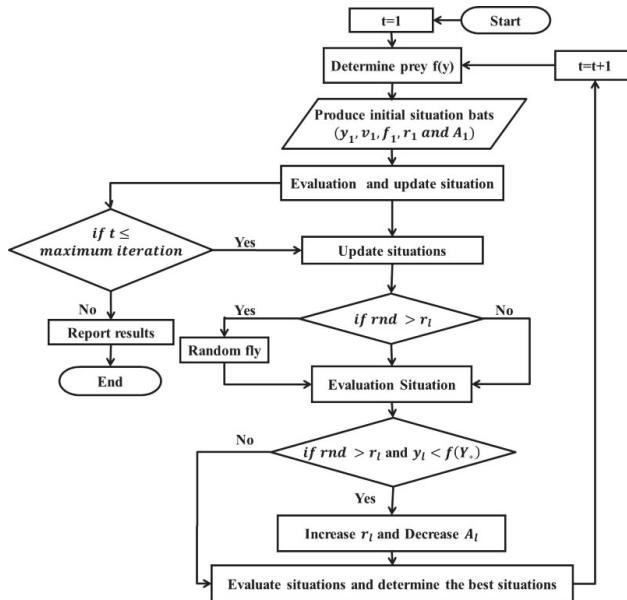
For a  $d$ -dimensional search space, the population is formed by  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ , and the velocity is given by  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ . The best previous position is given by  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})^T$ , and the g index is used for the global solution. The velocity and the position are updated, based on the following equations:

$$v_{id}^{n+1} = \chi \left[ w v_{id}^n + \frac{c_1 r_1 (p_{id}^n - x_{id}^n)}{\Delta t} + \frac{c_2 r_2 (p_{gd}^n - x_{id}^n)}{\Delta t} \right] \quad (14)$$

$$x_{id}^{n+1} = x_{id}^n + \Delta t v_{id}^{n+1} \quad (15)$$

where  $d$  is the number of dimensions,  $\chi$  is the constriction coefficient,  $N$  is the size of the swarm,  $w$  is the inertial weight,  $c_1$  and  $c_2$  are the acceleration coefficients,  $r_1$  and  $r_2$  are the random numbers,  $\Delta t$  is the time step,  $x_{id}^{n+1}$  is the new position,  $v_{id}^{n+1}$  is the new velocity vector, and  $n$  is time index.

First, the random population of the swarm is initialized, and then the objective function is computed to determine the local and global leaders. Finally, the velocity and the position are updated, and the cycle continues until the convergence criterion is satisfied.



**Figure 1.** Bat Algorithm procedure.

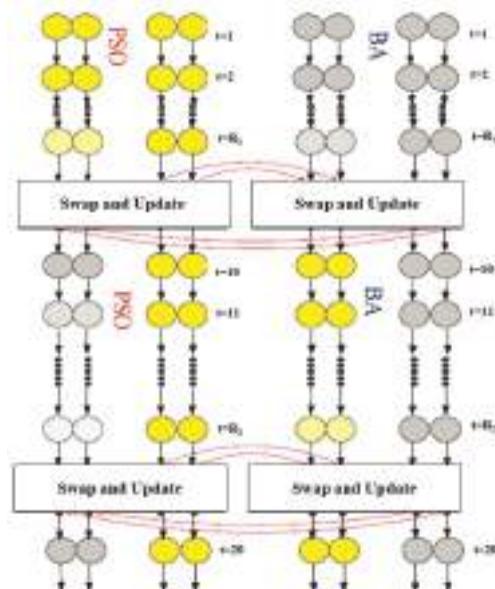
#### 2.4. Hybrid PSO and BA

The new HA acts based on a communication strategy between two algorithms. The idea of this strategy is that the parallel performance of the two algorithms allows the weaker solutions of one algorithm to be substituted with those of the other algorithm. The parallel structure has subgroups that are based on the division of the population. The subgroups act independently for each iteration. Thus, the objective function is computed for each group, and the weaker solutions of each group are selected and substituted with the better solutions of the subgroups of the other algorithm. The total number of iterations is  $R$ , and the total populations is  $N$ . Moreover,  $N_1$  and  $N_2$  are equal, i.e.,  $N/2$ . Figure 2 shows the performance of the HA.

The algorithm acts based on the following levels:

- (1) The random parameters for both algorithms (PSO + BA) are initialized, and the initial populations for the two algorithms are considered.
- (2) The first initial values for the hydrological parameters ( $K$ ,  $x$ ,  $m$  and  $\alpha$ ) are considered at the start of the algorithm.
- (3) The variation in storage is computed based on Equation (7). The initial outflow is the same as inflow.
- (4) The accumulated storage is computed based on Equation (8).
- (5) The outflow is computed based on Equation (6).
- (6) The time step is compared with the total flood time. If it is less than the total time, the algorithm goes to step 3; otherwise, the algorithm goes to the next level.
- (7) The objective function is computed for the two algorithms and all members that can be seen in the algorithms.

- (8) The velocity and position for the PSO algorithm are updated based on Equations (14) and (15), and the velocity, frequency and position are updated based on Equations (9)–(11).
- (9) The best particles migrate from the PSO algorithm to the BA, and there is a condition for BA similarity. In fact, the specific number of best members for each algorithm is known and is substituted for the worst solutions of the other algorithm.
- (10) The convergence criteria are considered. If the criteria are satisfied, the algorithm finishes; otherwise, the algorithm returns to the second step.



**Figure 2.** Hybrid algorithm procedure.

The following indexes are used for comparison of the different methods [35–38]:

- (1) The sum of the squared deviations between observed and estimated discharges is considered the objective function and is computed based on the following equation:

$$\text{Minimize}(SSQ) = \sum_{t=1}^n (O_{st} - O_{bt})^2 \quad (16)$$

- (2) The SAD between estimated and observed discharges is computed based on the following equation:

$$\text{Minimize}(SAD) = \sum_{t=1}^n |O_{bt} - O_{st}| \quad (17)$$

- (3) The mean absolute error (MARE) between estimated and observed discharges is computed based on the following equation:

$$MARE = \frac{1}{N} \sum_{i=1}^N \frac{|O_{bt} - O_{st}|}{O_{bt}} \quad (18)$$

(4) The error for peak discharge (EO) is computed based on the following equation:

$$EO = \frac{|O_{peak,bt} - O_{peak,st}|}{O_{peak,bt}} \quad (19)$$

(5) The error for peak time is computed based on the following equation:

$$ET = \left| T_{peak,bt} - T_{peak,st} \right| \quad (20)$$

where  $O_{st}$  is the simulated discharge,  $O_{bt}$  is the observed discharge,  $T_{peak,bt}$  is the peak time of observed discharge,  $T_{peak,st}$  is the peak time of simulated discharge, and N is the number of data.

### 3. Case Studies

In this section, brief introduction on the selected case studies have been reported. The first one is the Wilson flood event. The Wilson flood was selected for comparative analysis because it has been widely studied in the literature, resulting in a comprehensive body of information for comparison between the new hybrid method and other methods. This is a benchmark experimental problem that was considered by Wilson [33–37]. The time flood is equal to 120 h, and the peak occurs at a time step of 60 h with a value of 85 cm.

On the other hand, the second case study is the Karahan flood event. The Karahan flood, known as a benchmark problem in flood routing, is based on the 1960 flood of the Wye River in the United Kingdom [14,33–36]. The Wye River from Everwood to Belmont does not have any tributaries, and it has a small lateral flow, which is considered an important problem for flood routing. The flood time is 198 h, and the peak time is at the step time of 102 h.

Finally, the third case study is the Viessman and Lewis flood event. The Viessman and Lewis [39] multi-peak flood hydrograph was also selected as a benchmark study in this research.

## 4. Results and Discussion

### 4.1. Wilson Flood

#### 4.1.1. Sensitivity Analysis for Different Algorithms for the Wilson Flood

Tables 1–3 show the inflow and outflow values calculated by different methods. The evolutionary algorithm parameters do not have specific values at the start of the algorithm. Thus, sensitivity analysis was used to obtain the parameters. The objective function selected for this study was SSQ, and the variation in the objective function value was computed for various values of parameters. The best value for each parameter was selected when the objective function value was the minimum. For example, the population size for the HA varied from 20 to 80. When the size was 60, the SSQ value was 4.234 and led to the lowest objective function value. Further, the maximum frequency varied from 3 to 9. When the maximum frequency was 7 Hz, the objective function had the best value. The minimum frequency varied from 0.1 to 0.4, and the best value for the minimum frequency was 0.2. In addition, the maximum loudness varied from 0.2 to 0.8, and the best value based on the objective function value was 0.6 dB. The other parameters are listed in Table 1. For example, the acceleration coefficient varied from 1.6 to 2.2. The best value for this parameter was 2, because the SSQ, which was 4.233, was the lowest. Further, the inertia coefficient varied from 0.3 to 0.9. The best value was 0.7 because the SSQ was 4.234.

**Table 1.** Sensitivity analysis for hybrid algorithm.

Population Size	Objective Function	Maximum Frequency (Hz)	Objective Function	Minimum Frequency (Hz)	Objective Function	Maximum Loudness (dB)	Objective Function	$c_1$	Objective Function	$c_2$	Objective Function	$w$	Objective Function
20	5.123	3	5.254	1	5.565	0.20	4.999	1.6	5.145	1.6	5.011	0.3	5.133
40	4.789	5	4.884	2	4.987	0.40	4.845	1.8	4.933	1.8	4.987	0.5	4.654
60	4.234	7	4.233	3	4.234	0.60	4.234	2.0	4.234	2.0	4.234	0.70	4.235
80	4.312	9	4.679	4	4.789	0.80	4.565	2.2	4.555	2.2	4.445	0.90	4.512

**Table 2.** Sensitivity analysis for PSO.

Population Size	Objective Function	$c_1$	Objective Function	$c_2$	Objective Function	$w$	Objective Function
20	5.981	1.60	5.891	1.60	5.954	0.30	5.845
40	5.785	1.80	5.654	1.80	5.878	0.50	5.764
60	5.555	2.0	5.554	2.0	5.554	0.70	5.525
70	5.894	2.2	5.892	2.2	5.891	0.90	5.789

**Table 3.** Sensitivity analysis for BA.

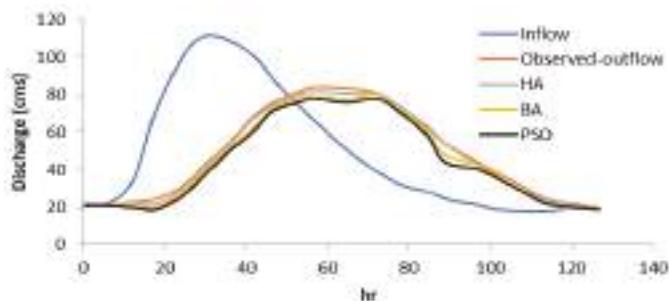
Population Size	Objective Function	Maximum Frequency (Hz)	Objective Function	Minimum Frequency (Hz)	Objective Function	Loudness (dB)	Objective Function
20	5.765	3	5.812	1	5.911	0.3	5.912
40	5.455	5	5.691	2	5.783	0.5	5.678
60	5.342	7	5.342	3	5.343	0.70	5.343
70	5.694	9	5.611	4	5.455	0.90	5.678

#### 4.1.2. Ten Random Results for Different Algorithms for Wilson Flood

Table 4 shows 10 random results for different algorithms. The average solution for the HA was 4.234, while the average solutions were 5.554 and 5.342 for the PSO algorithm and BA, respectively. Thus, the objective function or error value decreased by 23.71% and 20.7% compared to those of the PSO algorithm and BA, respectively. The computational time for the HA was 20 s, while it was 27 s and 25 s for the PSO algorithm and BA; thus, the computational time decreased by 25% and 20% in the HA compared to those in the PSO algorithm and BA, respectively. Additionally, the coefficient of variation for the HA was smaller than that for the PSO algorithm and BA and was therefore nonsignificant, which implied that the one-run program for the HA was reliable. Figure 3 shows the convergence method for the three algorithms, and it is apparent that the HA converged faster than the PSO algorithm and BA did. In fact, the researchers present the 10 or 15 random solutions to show the variation of solutions. It is apparent that all runs for the HA are close to the average solution and that all runs are reliable as the final solution. Although the BA and PSO have the solutions that are closest to the average solution, the other factors are important for the selection of the best methods. Table 4 shows that the hybrid method based on soft computing is the best method because the minimum, maximum and average solutions for the 10 results are close to each other. Additionally, the computation time for the HA is less than other methods (based on a PC with an i5 2.4 GHz CPU and 500 GB RAM), and the average of the error objective function based on HA is less than those of the other two methods.

**Table 4.** Ten random results for average solutions for Wilson flood.

Run Number	HA	BA	PSO
1	4.234	5.342	5.555
2	4.233	5.348	5.555
3	4.234	5.342	5.555
4	4.234	5.342	5.555
5	4.234	5.342	5.559
6	4.233	5.342	5.560
7	4.234	5.342	5.555
8	4.234	5.342	5.555
9	4.234	5.342	5.555
10	4.234	5.342	5.555
Average	4.234	5.342	5.555
Computational time	20 s	27 s	25 s
Variation coefficient	0.00007	0.0003	0.0004



**Figure 3.** Simulated discharges based on different methods for the Wilson flood.

#### 4.1.3. Discussion of the Wilson Flood Results

Table 5 shows the performance of the HA, PSO algorithm and BA based on the 4PMM (four-parameter Muskingum model). The computed SSQ based on the HA was 4.234, and the HA

method decreased the SSQ by 23.7% and 20.74% compared to those of the PSO and BA methods, respectively. The SAD value based on the HA decreased by 25% and 24% compared to those of the PSO algorithm and BA, respectively. The error between the simulated discharge and the observed discharge was shown by the EO, and the minimum value, 0.0111, was related to the HA. Additionally, the three methods predicted the peak time well, with an ET equal to zero. Table 6 compares the performances of different models based on the current research and previous studies. The performance of the HA based on the 4PMM was better than that based on the 3PMM because the error index for the 4PMM HA was lower than that for the 3PMM HA. For example, SSQ and SAD for the 4PMM were 4.234 and 3.125, respectively, while they were 12.25 and 10.95 for the 3PMM. Table 6 shows the performance of GA, HS, and the imperial competitive algorithm (ICA) based on the literature. The highest value of SSQ was related to GA based on the 3PMM; thus, this model had the worst performance. The SAD for the ICA was 23.46, which indicated that this model had the worst performance based on SAD and 3PMM (three-parameter Muskingum model). When the performances of the BA and PSO algorithm based on the 3PMM were compared with that based on the 4PMM, the models based on the 4PMM had the best performance because of their smallest index errors. All the models in Table 6 predicted the peak time well, with an ET value equal to zero. Table 7 shows the inflow and outflow and the peak value for the observed discharge, which was 85 cm when the time was 60 h. The results showed that peak discharge based on the HA was closer to the observed value. Table 8 shows the values of different coefficients for the Muskingum models. Generally, the results showed that the HA based on the 4PMM performed better than the other models did, with an SSQ value that decreased by 65%, 72% and 47% compared to those of the BA (3PMM), PSO algorithm (3PMM) and HA (3PMM), respectively. The values for other indexes supported this trend. Figure 3 shows that the HA based on the 4PMM performed better than the PSO and BA based on the 4PMM during the flood. Additionally, the comparison of results with other research shows the superiority of HA compared to the other algorithms. The honey bee mating algorithm was considered for flood routing [34]. SSQ based on 3PMM was 37.451, while the 3PMM and HA have smaller values for SSQ. Thus, HA has a better performance than HBMO.

Thus, all sections for the HA show the superiority of the method compared to the other methods. Although, the Muskingum method based on four parameters has more parameters than the Muskingum model based on three parameters, it is important to have a simulated hydrograph that matches with the observed flood. This section shows that prediction of peak discharge is the first priority in flood hydrographs. The observed peak in the hydrograph is 85 cm, while the simulated peak discharge is 85.011 cm. Clearly, the new method has good performance for this issue. The method is the best method for flood routing among other evolutionary algorithms because the peak time can be computed accurately. Table 5 shows the peak time predicted based on HA so that there is no delay between the computed time based on HA and the real time. The flood-emergency management authorities can make the best decisions based on the importance of the different projects.

**Table 5.** Comparison of results based on four-parameter Muskingum model for Wilson flood.

Method	SSQ	SAD	MARE	EO	ET
HA	4.234	3.125	0.012	0.00111	0
PSO	5.555	4.128	0.017	0.00251	0
BA	5.342	4.117	0.015	0.00167	0

**Table 6.** Comparison of results for Wilson flood.

Method	SSQ	SAD	MARE	EO	ET
GA [40] (Three-parameter Muskingum)	38.230	23.00	0.0912	0.0083	0
HS [40] (Three-parameter Muskingum)	36.780	23.40	0.0878	0.0107	0
ICA [40] (Three-parameter Muskingum)	36.801	23.46	0.0745	0.0105	0
BA (current research) (Three-parameter Muskingum)	12.25	10.95	0.0215	0.0079	0
PSO (current research) (Three-parameter Muskingum)	14.78	12.72	0.0325	0.0081	0
HA (current research) (Three-parameter Muskingum)	8.215	6.515	0.0205	0.0043	0

**Table 7.** Inflow and outflow for flood routing.

Time	Inflow (cm)	Outflow (Observed-cm)	Hybrid Method (cm)	BA (cm)	PSO
0	22	22	22.0	22.0	22.0
6	23	21	22.0	23.0	23.0
12	35	21	21.0	22.5	23.5
18	71	26	25.0	25.0	26.0
24	103	34	34.0	35.0	35.5
30	111	44	43.5	44.0	44.0
36	109	55	54.0	55.0	55.5
42	100	66	66.0	67.0	68.0
48	86	75	74.0	74.0	75.0
54	71	82	81.5	82.0	83.0
60	59	85	85.0011	85.00251	85.00267
66	47	84	84.0	84.0	84.0
72	39	80	81.0	80.5	81.0
78	32	73	74.0	73.0	74.0
84	28	64	64.0	65.0	66.0
90	24	54	54.0	55.0	56.0
96	22	44	44.0	44.0	45.0
102	21	36	36.0	37.0	38.0
108	20	30	30.5	31.0	31.0
114	19	25	25.5	26.2	26.9
120	19	22	23.0	24.0	25.0
126	18	19	20.0	21.0	22.0

**Table 8.** Extracted coefficients for the four-parameter Muskingum model.

Method	K	x	m	$\alpha$
HA	0.164	0.2879	3.781	0.4678
BA	0.152	0.2768	3.567	0.4567
PSO	0.144	0.2645	3.123	0.3789

#### 4.2. Karahan Flood

##### 4.2.1. Discussion of the Karahan Results

The parameters used for the HA were maximum frequency, minimum frequency, maximum loudness, initial population, acceleration coefficient, and inertia coefficient; which were computed based on sensitivity analysis described in the previous section (maximum frequency: 7; minimum

frequency: 3; maximum loudness: 0.95; initial population: 50; acceleration coefficient: 2; and inertial weight: 0.7).

Table 9 compares the performances of different algorithms based on the 4PMM for the Karahan flood. The SSQ of the HA and the 4PMM was 30,235, and the SSQ of the HA decreased by 5.85% and 2.8% compared to those of the PSO algorithm and BA, respectively. The SAD for the HA based on the 4PMM was 625, which were 0.32% and 7.54% lower than those of the PSO algorithm and BA, respectively. The difference between the simulated peak discharge and the observed discharge was nonsignificant; the EO of the HA equaled 0.101, a value that was 7.3% and 6.4% lower than those of the PSO algorithm and BA. The best performance based on MARE was related to the HA, which produced the lowest MARE value. Additionally, the three methods predicted the peak discharge well, with very low EO values. The GA had the worst performance of the methods, as shown in Table 10. The SSQ for the HS method was 39,944, which was the highest SSQ value produced by the models based on the 4PMM and 3PMM. The HA based on the 3PMM performed better than the other methods based on the 3PMM, as the SSQ of the HA decreased by 11.38%, 18%, 7.1%, 3.4% and 6.3% compared to those of the GA, HS, ICA, BA and PSO algorithm, respectively. The HA based on the 3PMM produced the lowest SAD value of all the 3PMM-based models. The performances of the HA, PSO algorithm and BA based on the 4PMM were comparable to the corresponding performances based on the 3PMM, but the 4PMM algorithm performed better than did the 3PMM algorithm. For example, the SSQ values of the HA, PSO algorithm and BA based on the 4PMM were 2.8%, 3.4% and 0.27% lower than those for the HA, PSO algorithm and BA based on the 3PMM, respectively. Table 11 shows the inflow and outflow for different methods and the peak discharge, which was 830 cm when the step time was 96 h. These results show that the discharges estimated by the HA were near to the observed discharges. Table 12 shows the extracted coefficients for different algorithms. Figure 4 shows the HA based on the 4PMM had the best performance during the flood, with a good match between the observed discharges and the discharges estimated by the HA. The general results for this section show that the HA based on 3PMM and 4PMM can predict the peak time well such that the ET parameter for HA based on 3PMM and 4PMM equals zero. Another point is related to the value of predicted discharge such that the EO index shows the ability of the different methods. 4PMM models based on HA, PSO and BA have a better performance: If the EO based on the table equals to 0.701, the EO based on Table 9 and FPMM models is less than 0.701. Although the 3PMM needs less parameters, the time and value peak can be predicted accurately based on 4PMM. Clearly, the importance of the model accuracy to decision makers can be considered in the selection of 3PMM and 4PMM.

**Table 9.** Comparison of results based on the four-parameter Muskingum model for Karahan flood.

Method	SSQ	SAD	MARE	EO	ET
HA	30,235	625	0.22	0.101	0
PSO	32,119	697	0.25	0.109	0
BA	31,112	676	0.24	0.108	0

**Table 10.** Comparison of results for Karahan flood.

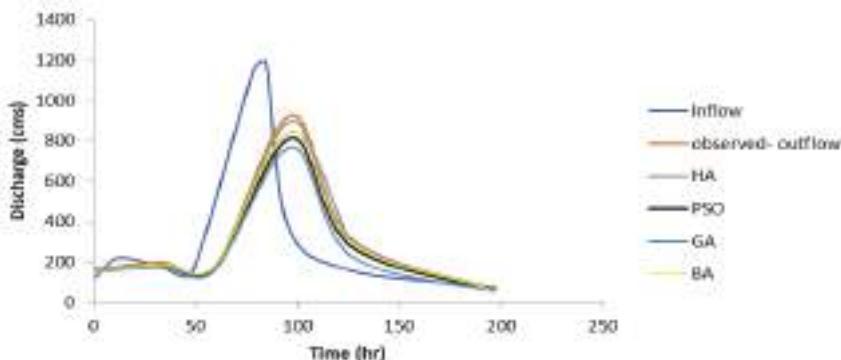
Method	SSQ	SAD	MARE	EO	ET
GA [40] (Three-parameter Muskingum)	35,123	1980	0.910	0.701	0
HS [40] (Three-parameter Muskingum)	37,944	2161	0.924	0.798	0
ICA [40] (Three-parameter Muskingum)	37,825	2054	0.914	0.787	0
BA (current research) (Three-parameter Muskingum)	32,228	712	0.420	0.115	0
PSO (current research) (Three-parameter Muskingum)	33,229	735	0.454	0.125	0
HA (current research) (Three-parameter Muskingum)	31,125	697	0.254	0.105	0

**Table 11.** The inflow and outflow for flood routing.

Time	Inflow (cm)	Outflow (Observed-cm)	Hybrid Method (cm)	BA (cm)	PSO (cm)
0	154	102	102.0	102.0	102.0
6	150	140	139.23	138.23	154.2
12	219	169	170.21	171.24	152.1
18	182	190	185.12	183.24	179.4
24	182	209	202.34	200.11	190.9
30	192	218	212.23	198.23	185.4
36	165	210	207.11	192.32	186.9
42	150	194	192.12	189.23	180.2
48	128	172	170.21	169.24	164.1
54	168	149	147.21	146.74	143.7
60	260	136	137.21	139.23	152.8
66	471	228	219.21	212.23	196.3
72	717	303	300.11	298.21	267.3
78	1092	366	358.11	354.23	351.4
84	1145	456	436.32	426.73	431.8
90	600	615	612.21	623.24	617.4
96	365	830	830.101	830.108	830.109
102	277	969	894.12	879.12	836.70
108	227	665	665.101	665.108	665.109
114	187	519	519.21	523.12	549.10
120	161	444	435.68	424.32	416.90
126	143	321	315.23	312.11	305.0
132	126	208	210.21	212.21	221.40
138	115	176	169.21	166.24	163.38
144	102	148	142.12	139.23	131.20
150	93	125	119.21	115.67	110.0
156	88	114	109.21	100.21	96.40
162	82	106	110.21	112.11	89.20
168	76	97	92.21	89.23	82.70
174	73	89	82.12	79.43	76.30
180	70	81	80.23	78.12	73.00
186	67	76	79.14	75.12	69.80
192	63	71	70.14	70.11	66.7
198	59	66	70.23	69.12	62.40

**Table 12.** The extracted coefficients for the four-parameter Muskingum model.

Method	<i>K</i>	<i>x</i>	<i>m</i>	<i>α</i>
HA	0.610	0.404	3.781	1.125
BA	0.578	0.311	2.896	1.112
PSO	0.578	0.309	2.789	1.105

**Figure 4.** Simulated discharges by different algorithms for Karahan flood.

#### 4.2.2. Ten Random Results for Karahan Flood

Table 13 shows the averages of 10 random results for different methods. The average solution of the HA was lower than that of the PSO algorithm and BA. Additionally, the computational time for the HA was 17% and 29% lower than those for the BA and PSO algorithm, respectively. Additionally, the coefficient of variation for the HA was a small value, which proved that the HA based on one computer program run can be reliable, producing high-quality solutions. Table 13 helps determine the best decision based on time, value of objective function and variation coefficient. The least value for the error objective function is provided by the HA based on the least value of the error objective function. Additionally, the least probable time is related to the HA, and all the runs for the HA have the smallest probable variations. Thus, the method encourages the decision makers to select HA based on the mentioned indexes. Additionally, Table 14 shows the sensitivity analysis for the HA and 4PMM model. The best population size for the HA is 60, and the best value for the maximum frequency and minimum frequency is 7 and 3. Other parameters can be seen in Table 14.

**Table 13.** Investigation of different methods for Karahan flood.

Run Number	HA	BA	PSO
1	30,235	31,112	32,119
2	30,237	31,117	32,119
3	30,235	31,112	32,119
4	30,235	31,112	32,119
5	30,235	31,112	32,119
6	30,235	31,112	32,119
7	30,237	31,112	32,122
8	30,235	31,117	32,112
9	30,235	31,112	32,119
10	30,235	31,112	32,119
Average	30,235.4	31,113	32,119
Computational time	19 s	23 s	27 s
Variation coefficient	0.00002	0.00005	0.00007

#### 4.3. Discussion of the Viessman and Lewis Flood Results

The Viessman and Lewis Fi [39] multi-peak flood hydrograph was selected for this section. Table 15 shows the performances of different models in analysing the Viessman and Lewis flood. The results were compared with those of the WA in the literature, and this is a synthetic problem [33–37]. The highest SSQ for this flood, 73,312, was produced by the WA based on the 3PMM. The HA based on the 4PMM performed better than did the other models. The SSQ, SAD and MARE values of the HA based on the 4PMM had the lowest values. The BA based on the 4PMM performed better than did the WA and PSO algorithm based on the 3PMM. For example, the SSQ of the BA based on the 4PMM was 47,224, which was 14%, 3.8%, 16.72% and 35% lower than those of the PSO algorithm (4PMM), BA (3PMM), PSO algorithm (3PMM) and WA (3PMM), respectively. Additionally, the HA, PSO algorithm and BA based on the 4PMM outperformed the PSO algorithm, BA and HA based on the 3PMM. Figure 5 shows that the estimated discharge of the HA based on the 4PMM matched the observed discharge during the flood well. Furthermore, Table 16 shows the sensitivity analysis for the 4PMM and HA. The best size population for the HA is 60 because it has the least value for the objective function. The maximum loudness for the HA is 0.60, and the other parameters can be seen in Table 16. The comparison of results with other research studies shows the superiority of 4PMM and HA. For example, one study considered flood routing based on improved PSO based on correction of Wight inertia and a Muskingum flood with two parameters. The MARE index is 0.911, while the MARE for the HA and 4PMM is 0.794. HA acts better than the improved PSO and the Muskingum model with two parameters [34].

Moreover, Table 17 shows the 10 random results for different methods, and it can be seen that the value of the error objective function for the 4PMM and HA is less than PSO and BA. Additionally, the computational time based on HA and 4PMM is less than those of other methods. The solution for the HA is high quality because the variation coefficient is small compared to those of the other methods.

**Table 14.** Sensitivity analysis for Karahan flood.

Population Size	Objective Function	Maximum Frequency (Hz)	Objective Function	Minimum Frequency (Hz)	Objective Function	Maximum Loudness (dB)	Objective Function	$c_1$	Objective Function	$c_2$	Objective Function	$w$	Objective Function
20	34,231	3	33,278	1	32,278	0.20	31,124	1.6	32,112	1.6	32,114	0.3	31,127
40	32,278	5	32,211	2	31,112	0.40	30,298	1.8	31,214	1.8	31,289	0.5	31,119
60	30,235	7	30,235	3	30,235	0.60	30,235	2.0	30,235	2.0	30,235	0.70	30,235
80	31,112	9	31,265	4	31,112	0.80	30,236	2.2	31,112	2.2	31,112	0.90	30,254

**Table 15.** Evaluation of different methods for Viessman and Lewis flood.

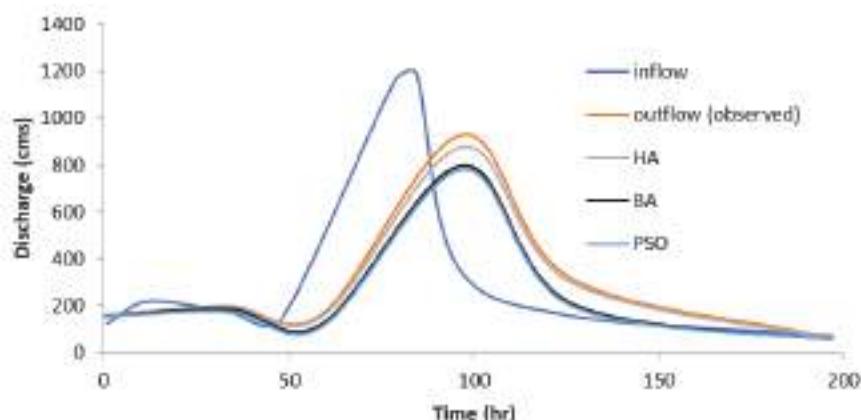
Method	SSQ	SAD	MARE	EO	ET
HA (4PMM)	45,225	998.24	0.794	0.111	0
PSO (4PMM)	55,124	1012.22	0.812	0.209	0
BA (4PMM)	47,224	1001.14	0.798	0.118	0
HA (3PMM)	48,225	1002.23	0.812	0.115	0
PSO (3PMM)	56,712	1014.45	0.867	0.288	0
BA (3PMM)	49,112	1009.23	0.724	0.202	0
WA (3PMM)	73,312	1037.25	0.994	0.488	0

**Table 16.** Sensitivity analysis for Viessman and Lewis flood.

Population Size	Objective Function	Maximum Frequency (Hz)	Objective Function	Minimum Frequency (Hz)	Objective Function	Maximum Loudness (dB)	Objective Function	$c_1$	Objective Function	$c_2$	Objective Function	$w$	Objective Function
20	47,229	3	47,312	1	49,278	0.20	46,124	1.6	47,119	1.6	48,124	0.3	48,119
40	46,214	5	47,001	2	47,112	0.40	45,298	1.8	46,224	1.8	47,211	0.5	47,015
60	45,225	7	45,225	3	45,225	0.60	45,225	2.0	45,225	2.0	45,225	0.70	45,225
80	49,112	9	45,287	4	48,112	0.80	47,119	2.2	46,179	2.2	46,117	0.90	46,119

**Table 17.** Investigation of different methods for Karahan flood.

Run Number	HA	PSO	BA
1	45,225	55,124	47,224
2	45,226	55,124	47,226
3	45,225	55,127	47,224
4	45,225	55,124	47,224
5	45,225	55,124	47,224
6	45,225	55,124	47,224
7	45,225	55,124	47,224
8	45,225	55,124	47,224
9	45,225	55,124	47,224
10	45,225	55,124	47,224
Average	45,225	31113	47,224
Computational time	15 s	17 s	19 s
Variation coefficient	0.000004	0.000006	0.000005

**Figure 5.** Extracted hydrograph for Viessman and Lewis flood based on 4PMM.

## 5. Conclusions

The present study considers flood routing based on the new hybrid BA and PSO algorithm. The new algorithm is based on the substitution of the weaker BA solutions with PSO. The structure of the new HA, with the elimination of the weaker BA solutions, enables the escape from local optimums. Three case studies that are benchmark studies in the flood routing field were used in this study, and the 4PMM was selected for analysis. For the Wilson flood case study, the computational time of the HA method was 25% and 20% lower than those of the PSO algorithm and BA methods, respectively. The SSQ computed by the HA was 4.234, which was 23.7% and 20.74% lower than those computed by the PSO algorithm and BA, respectively. Generally, the Wilson flood results showed that the HA based on the 4PMM performed better than the other models, with an SSQ value 65%, 72% and 47% lower than those of the BA (3PMM), PSO algorithm (3PMM) and HA (3PMM), respectively; the values of the other indexes agreed with this result. The results for the Karahan flood case study showed that the performances of different models based on the BA, PSO algorithm and HA with the 4PMM were better than those of the same models with the 3PMM. In the Karahan flood case study, the HA based on the 3PMM performed better than the other methods based on the 3PMM did, with an SSQ value that was 11.38%, 18%, 7.1%, 3.4%, and 6.3% lower than those of the GA, HS, ICA, BA and PSO methods, respectively. For the Viessman and Lewis flood case study, the SSQ, SAD and MARE of the HA based

on the 4PMM were the lowest. The HA based on the 4PMM performed better than did the BA and PSO based on the 3PMM. The results indicated that the HA based on substitution of weaker solutions of each algorithm with the strong solutions can decrease the computational time, and the chance of obtaining the best solutions increased significantly. With the new strategy, there is no problem with trapping in the local optimums because the elimination of weaker solutions can lead to exits from the local optimums.

However, the HA based on the 4PMM had the best performance of all the models, suggesting that it should be used in future studies with other advanced Muskingum models and more parameters, such as 7- or 9-parameter Muskingum models, to evaluate the skill of HA for flood routing prediction modelling.

**Author Contributions:** A.E.-S. and V.P.S. initiated the research point of the hydrological problem and supervised the study. M.E. collected the data and performed the modeling. M.B.A.M., A.N.A. and M.F.A. handled the writing up of the introduction and methodology. The analysis of the modeling outcomes have been handled by F.B.O., S.S., Z.M.Y., H.A.A. and M.E.; Z.M.Y. and H.A.A. organized the whole manuscript and managed the paper submission and revision.

**Funding:** This research was funded by University of Malaya research grants RP017C-15SUS and RP025A-18SUS.

**Acknowledgments:** The authors appreciate the technical and financial support received from the University of Malaya.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bagatur, T.; Onen, F. Development of predictive model for flood routing using genetic expression programming. *J. Flood Risk Manag.* **2018**, *11*, S444–S454. [[CrossRef](#)]
2. El-shafie, A.; Noureldin, A.; Taha, M.; Hussain, A.; Mukhlisin, M. Dynamic versus static neural network model for rainfall forecasting at Klang River Basin, Malaysia. *Hydrol. Earth Syst. Sci.* **2012**, *16*, 1151–1169. [[CrossRef](#)]
3. Mazzoleni, M.; Noh, S.J.; Lee, H.; Liu, Y.; Seo, D.J.; Amaranto, A.; Alfonso, L.; Solomatine, D.P. Real-time assimilation of streamflow observations into a hydrological routing model: Effects of model structures and updating methods. *Hydrol. Sci. J.* **2018**, *63*, 386–407. [[CrossRef](#)]
4. Zhou, Y.; Guo, S.; Chang, F.J.; Liu, P.; Chen, A.B. Methodology that improves water utilization and hydropower generation without increasing flood risk in mega cascade reservoirs. *Energy* **2018**, *143*, 785–796. [[CrossRef](#)]
5. El-shafie, A.; Mukhlisin, M.; Najah, A.A.; Taha, M.R. Performance of artificial neural network and regression techniques for rainfall-runoff prediction. *Int. J. Phys. Sci.* **2011**, *6*, 1997–2003. [[CrossRef](#)]
6. Vatankhah, A.R. Discussion of “Assessment of Modified Honey Bee Mating Optimization for Parameter Estimation of Nonlinear Muskingum Models” by Majid Niazkar and Seied Hosein Afzali. *J. Hydrol. Eng.* **2018**, *23*, 07018002. [[CrossRef](#)]
7. Yaseen, Z.M.; El-shafie, A.; Jaafar, O.; Afan, H.A.; Sayl, K.N. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *J. Hydrol.* **2015**, *530*, 829–844. [[CrossRef](#)]
8. Pati, A.; Sen, S.; Perumal, M. Modified Channel-Routing Scheme for SWAT Model. *J. Hydrol. Eng.* **2018**, *23*, 1–8. [[CrossRef](#)]
9. Yaseen, Z.M.; Jaafar, O.; Deo, R.C.; Kisi, O.; Adamowski, J.; Quilty, J.; El-shafie, A. Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq. *J. Hydrol.* **2016**. [[CrossRef](#)]
10. Barati, R. Comment on “Evaluation of a physically based quasi-linear and a conceptually based nonlinear Muskingum methods” by Muthiah Perumal, Gokmen Tayfur, C. Madhusudana Rao and Gurhan Gurarslan. *J. Hydrol.* **2017**, *550*, 218–219. [[CrossRef](#)]
11. Barbetta, S.; Moramarco, T.; Perumal, M. A Muskingum-based methodology for river discharge estimation and rating curve development under significant lateral inflow conditions. *J. Hydrol.* **2017**, *554*, 216–232. [[CrossRef](#)]

12. Barati, R.; Badfar, M.; Azizyan, G.; Akbari, G.H. Discussion of “Parameter Estimation of Extended Nonlinear Muskingum Models with the Weed Optimization Algorithm” by Farzan Hamed, Omid Bozorg-Haddad, Maryam Pazoki, Hamid-Reza Asgari, Mehran Parsa and Hugo A. Loáiciga. *J. Irrig. Drain. Eng.* **2018**, *144*, 07017021. [[CrossRef](#)]
13. Yoo, C.; Lee, J.; Lee, M. Parameter Estimation of the Muskingum Channel Flood-Routing Model in Ungauged Channel Reaches. *J. Hydrol. Eng.* **2017**, *22*, 05017005. [[CrossRef](#)]
14. Zhang, S.; Kang, L.; Zhou, B. Parameter estimation of nonlinear Muskingum model with variable exponent using adaptive genetic algorithm. In *Environmental Conservation, Clean Water, Air & Soil (CleanWAS)*; IWA Publishing: London, UK, 2017; pp. 231–237.
15. Luo, J.; Xie, J. Parameter Estimation for Nonlinear Muskingum Model Based on Immune Clonal Selection Algorithm. *J. Hydrol. Eng.* **2010**, *15*, 844–851. [[CrossRef](#)]
16. Barati, R. Parameter Estimation of Nonlinear Muskingum Models Using Nelder-Mead Simplex Algorithm. *J. Hydrol. Eng.* **2011**, *16*, 946–954. [[CrossRef](#)]
17. Barati, R.; Rahimi, S.; Akbari, G.H. Analysis of dynamic wave model for flood routing in natural rivers. *Water Sci. Eng.* **2012**, *5*, 243–258. [[CrossRef](#)]
18. Karahan, H.; Gurarslan, G.; Geem, Z.W. Parameter Estimation of the Nonlinear Muskingum Flood-Routing Model Using a Hybrid Harmony Search Algorithm. *J. Hydrol. Eng.* **2013**, *18*, 352–360. [[CrossRef](#)]
19. Orouji, H.; Haddad, O.B.; Fallah-Mehdipour, E.; Mariño, M.A. Estimation of Muskingum parameter by meta-heuristic algorithms. In Proceedings of the Institution of Civil Engineers: Water Management; Institution of Civil Engineers (ICE): London, UK, 2013; Volume 166, pp. 315–324.
20. Easa, S.M.; Barati, R.; Shahheydari, H.; Nodoshan, E.J.; Barati, T. Discussion: New and improved four-parameter non-linear Muskingum model. *Proc. Inst. Civ. Eng. Water Manag.* **2014**, *167*, 612–615. [[CrossRef](#)]
21. Ouyang, A.; Li, K.; Truong, T.K.; Sallam, A.; Sha, E.H.M. Hybrid particle swarm optimization for parameter estimation of Muskingum model. *Neural Comput. Appl.* **2014**, *25*, 1785–1799. [[CrossRef](#)]
22. Niazkar, M.; Afzali, S.H. Assessment of modified honey bee mating optimization for parameter estimation of nonlinear Muskingum models. *J. Hydrol. Eng.* **2015**, *20*, 04014055. [[CrossRef](#)]
23. Talatahari, S.; Sheikholeslami, R.; Farahmand Azar, B.; Daneshpajouh, H. Optimal Parameter Estimation for Muskingum Model Using a CSS-PSO Method. *Adv. Mech. Eng.* **2013**, *5*, 480954. [[CrossRef](#)]
24. Karahan, H.; Gurarslan, G.; Geem, Z.W. A new nonlinear Muskingum flood routing model incorporating lateral flow. *Eng. Optim.* **2015**, *47*, 737–749. [[CrossRef](#)]
25. Bozorg Haddad, O.; Hamed, F.; Orouji, H.; Pazoki, M.; Loáiciga, H.A. A Re-Parameterized and Improved Nonlinear Muskingum Model for Flood Routing. *Water Resour. Manag.* **2015**, *29*, 3419–3440. [[CrossRef](#)]
26. Niazkar, M.; Afzali, S.H. Parameter estimation of new improved Muskingum model using new hybrid model. *Hydrol. Res.* **2016**, *49*, nh2016089. [[CrossRef](#)]
27. Dhar, S.; Kundu, M.K. A novel method for image thresholding using interval type-2 fuzzy set and Bat algorithm. *Appl. Soft Comput.* **2018**, *63*, 1–37. [[CrossRef](#)]
28. Cui, Z.; Cao, Y.; Cai, X.; Cai, J.; Chen, J. Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things. *J. Parallel Distrib. Comput.* **2018**. [[CrossRef](#)]
29. Bekdaş, G.; Nigdeli, S.M.; Yang, X.S. A novel bat algorithm based optimum tuning of mass dampers for improving the seismic safety of structures. *Eng. Struct.* **2018**, *159*, 89–98. [[CrossRef](#)]
30. Mihoubi, M.; Rahmouni, A.; Lorenz, P.; Lasla, N. An effective Bat algorithm for node localization in distributed wireless sensor network. *Secur. Priv.* **2017**, *1*, e7. [[CrossRef](#)]
31. Bozorg-Haddad, O.; Karimirad, I.; Seifollahi-Aghmuni, S.; Loáiciga, H.A. Development and Application of the Bat Algorithm for Optimizing the Operation of Reservoir Systems. *J. Water Resour. Plan. Manag.* **2015**, *141*, 04014097. [[CrossRef](#)]
32. Gill, M.A. Flood routing by the Muskingum method. *J. Hydrol.* **1978**, *36*, 353–363. [[CrossRef](#)]
33. Tung, Y. River Flood Routing by Nonlinear Muskingum Method. *J. Hydraul. Eng.* **1985**, *111*, 1447–1460. [[CrossRef](#)]
34. Geem, Z.; Kim, J.; Loganathan, G. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]

35. Elzwayie, A.; El-shafie, A.; Yaseen, Z.M.; Afan, H.A.; Allawi, M.F. RBFNN-based model for heavy metal prediction for different climatic and pollution conditions. *Neural Comput. Appl.* **2017**, *28*, 1991–2003. [[CrossRef](#)]
36. Yaseen, Z.M.; Fu, M.; Wang, C.; Hanna, W.; Wan, M.; Deo, R.C.; El-shafie, A. Application of the Hybrid Artificial Neural Network Coupled with Rolling Mechanism and Grey Model Algorithms for Streamflow Forecasting Over Multiple Time Horizons. *Water Res. Manag.* **2018**, *32*, 1883–1899. [[CrossRef](#)]
37. Diop, L.; Bodian, A.; Djaman, K.; Yaseen, Z.M.; Deo, R.C.; El-shafie, A.; Brown, L.C. The influence of climatic inputs on stream-flow pattern forecasting: Case study of Upper Senegal River. *Environ. Earth Sci.* **2018**, *77*, 182. [[CrossRef](#)]
38. Ghorbani, M.A.; Deo, R.C.; Yaseen, Z.M.; Kashani, M.H. Pan evaporation prediction using a hybrid multilayer perceptron-firefly algorithm (MLP-FFA) model: Case study in North Iran. *Theor. Appl. Climatol.* **2017**, *1–13*. [[CrossRef](#)]
39. Warren, V.; Gary, L.L. *Introduction to Hydrology*; Prentice Hall India (P) Ltd.: Delhi, India, 2003.
40. Moghaddam, A.; Behmanesh, J.; Farsijani, A. Parameters Estimation for the New Four-Parameter Nonlinear Muskingum Model Using the Particle Swarm Optimization. *Water Resour. Manag.* **2016**, *30*, 2143–2160. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Flood Hydrograph Prediction Using Machine Learning Methods

Gokmen Tayfur <sup>1,\*</sup>, Vijay P. Singh <sup>2</sup>, Tommaso Moramarco <sup>3</sup> and Silvia Barbetta <sup>3</sup>

<sup>1</sup> Department Civil Engineering, Izmir Institute of Technology, Urla, Izmir 35430, Turkey

<sup>2</sup> Department of Biological & Agricultural Engineering & Zachry Department of Civil Engineering, Texas A&M University, 321 Scoatess Hall, 2117 TAMU, College Station, TX 77843, USA; vsingh@tamu.edu

<sup>3</sup> IRPI, Consiglio Nazionale delle Ricerche, via Madonna Alta 126, 06128 Perugia, Italy; tommaso.moramarco@irpi.cnr.it (T.M.); s.barbetta@irpi.cnr.it (S.B.)

\* Correspondence: gokmentayfur@iyte.edu.tr; Tel.: +90-232-750-6815; Fax: +90-232-750-6801

Received: 28 June 2018; Accepted: 20 July 2018; Published: 24 July 2018

**Abstract:** Machine learning (soft) methods have a wide range of applications in many disciplines, including hydrology. The first application of these methods in hydrology started in the 1990s and have since been extensively employed. Flood hydrograph prediction is important in hydrology and is generally done using linear or nonlinear Muskingum (NLM) methods or the numerical solutions of St. Venant (SV) flow equations or their simplified forms. However, soft computing methods are also utilized. This study discusses the application of the artificial neural network (ANN), the genetic algorithm (GA), the ant colony optimization (ACO), and the particle swarm optimization (PSO) methods for flood hydrograph predictions. Flow field data recorded on an equipped reach of Tiber River, central Italy, are used for training the ANN and to find the optimal values of the parameters of the rating curve method (RCM) by the GA, ACO, and PSO methods. Real hydrographs are satisfactorily predicted by the methods with an error in peak discharge and time to peak not exceeding, on average, 4% and 1%, respectively. In addition, the parameters of the Nonlinear Muskingum Model (NMM) are optimized by the same methods for flood routing in an artificial channel. Flood hydrographs generated by the NMM are compared against those obtained by the numerical solutions of the St. Venant equations. Results reveal that the machine learning models (ANN, GA, ACO, and PSO) are powerful tools and can be gainfully employed for flood hydrograph prediction. They use less and easily measurable data and have no significant parameter estimation problem.

**Keywords:** machine learning methods; St. Venant equations; rating curve method; nonlinear Muskingum model; hydrograph predictions

## 1. Introduction

Flood routing in a river is important to trace the movement of a flood wave along a channel length and thereby calculate the flood hydrograph at any downstream section. This information is needed for designing flood control structures, channel improvements, navigation, and assessing flood effects [1].

There are basically two flood routing methods: (1) hydraulic methods that are based on numerical solutions of St. Venant equations or their simplified forms as the diffusion wave or the kinematic wave; and (2) hydrologic methods that are based solely on the conservation of mass principle [2], such as the Rating Curve Method (RCM) [3], Muskingum method (MM) or nonlinear Muskingum method (NMM) [4]. The above methods require substantial field data; such as cross-sectional surveying, roughness, flow depth and velocity measurements that are costly and time consuming. When lateral flow becomes significant, the flood prediction is affected by high uncertainty [5]. In addition, numerical

solutions of the St. Venant equations require a fair amount of data which is often not available and can encounter convergence and stability problems [2].

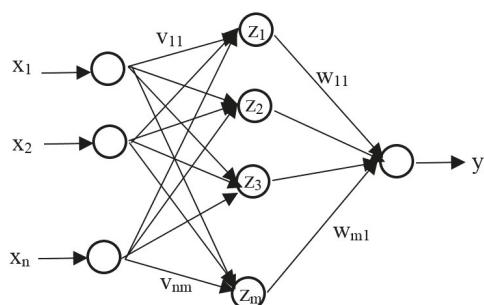
In the 1990s, the first applications of the artificial neural networks led to the realization that machine learning methods can handle nonlinear problems efficiently and satisfactorily without being restricted by the mathematical operations of integration and/or differentiation [6]. This even led to the development of new machine learning algorithms in the 1990s and the 2000s, such as harmony search (HS), gene expression programming (GEP), and genetic programming (GP). Machine learning (soft) methods are data driven and do not require substantial data and parameter estimation, as in the case of the distributed physically-based hydrologic models. The important advantages of these models are their relatively simple coding, low computational cost, fast convergence, and adaptiveness to new data [6,7].

Flood hydrograph predictions are performed using several soft computing methods, such as the artificial neural network (ANN) [8], the genetic algorithm (GA) [9,10], and the Fuzzy logic methods [11]. Most recently, the performance of some of the machine learning algorithms against the Variable Parameter Muskingum Model (VPMM) was compared [12]. In this context, it is of considerable interest to investigate how well the machine learning methods work in the real field and this study attempts to answer to the demand presenting real flood hydrograph predictions by the ANN, the GA, the ACO, and the PSO and compares the performances against those of the RCM. The flood hydrographs observed along an equipped branch of Tiber River, central Italy, with significant lateral flow contribution is used for testing the methods. In addition, the study tests the performance of these machine learning methods, based on the NMM, against that of the St. Venant equations by flood routing in an artificial channel having different bed slopes.

## 2. Material and Methods

### 2.1. Artificial Neural Network (ANN)

The artificial neural network is a processor made up of neurons. It can learn and store information through the training process. ANNs can handle nonlinearity and, therefore, have a wide application in many disciplines, including the flood hydrograph prediction in hydrology. In general, the three-layer network, as shown in Figure 1, is conceptualized in flood hydrograph prediction problems. In such network (see Figure 1); the compacted input values ( $x_i$ ) are first entered into the input layer neurons. They are multiplied by the connection weights ( $v_{ij}$ ) and then passed on to the hidden layer neurons. Each neuron sums all the received weighted information ( $x_i v_{ij}$ ) and then passes the sum through an activation function to produce an output ( $z_i$ ), which in turn, becomes the input signal for the output layer neuron. The output from each inner layer neuron is multiplied by the related connection weight ( $w_{ij}$ ) and then passed on to the output neuron, which sums all the received signal ( $z_i w_{ij}$ ) and passes it through an activation function to produce the network output ( $y_i$ ).



**Figure 1.** Three-layer network.

The optimal values of the connection weights can be found by the back propagation algorithm, which minimizes the error function by using the gradient decent method. The minimized error function,  $E$ , is expressed by Equation (1), as follows [11]:

$$E = \sum_i^p (t_i - y_i)^2 \quad (1)$$

where  $y_i$  is the network output,  $t_i$  is the user-specified target output, and  $p$  is the number of training patterns.

The connection weights ( $w_{ij}$ ) are updated at each iteration using Equation (2), as follows [11]:

$$\Delta w_{ij}(n) = \alpha \Delta w_{ij}(n-1) - \delta \frac{\partial E}{\partial w_{ij}} \quad (2)$$

where  $\Delta w_{ij}(n) = w_{ij}^{old} - w_{ij}^{new}$  at the present iteration ( $n$ ), and  $\Delta w_{ij}(n-1) = w_{ij}^{old} - w_{ij}^{new}$  at the previous iteration ( $n-1$ ).  $\delta$  is the learning rate and  $\alpha$  is the momentum factor, taking values between 0 and 1. In this study, tangent hyperbolic ( $\tanh$ ) activation function (Equation (3)) was employed.

$$f(sum) = \frac{2}{1 + e^{-2sum}} - 1 \quad (3)$$

where  $sum$  is the total information received by the neuron. The tangent hyperbolic function is bounded in between  $-1$  and  $+1$  and, therefore, all the values were compacted by Equation (4) into  $-0.9$  and  $+0.9$ .

$$z_i = \left[ \frac{1.8(x_i - x_{min})}{x_{max} - x_{min}} \right] - 0.9 \quad (4)$$

where  $z_i$  is the standardized value,  $x_{min}$  is the minimum value in the set; and  $x_{max}$  is the maximum value in the set. Details of ANNs are available in the literature [11].

## 2.2. Genetic Algorithm (GA)

Genetic algorithm (GA) was developed by Holland [13]. The concept is based on the survival of the fittest. It employs chromosomes, each of which consists of many genes. Each gene stands for a decision variable (or a model parameter) and each chromosome stands for a possible optimal solution. Note that each gene is formed from a string of many 0 and 1 digits. The fitness of a chromosome,  $F(C_i)$ , in a gene pool, which can contain many chromosomes, is found by Equation (5):

$$F(C_i) = \frac{f(C_i)}{\sum_{i=1}^N f(C_i)} \quad (5)$$

where  $N$  is the total number of chromosomes,  $C_i$  is the  $i^{th}$  chromosome,  $f(C_i)$  is the functional value of the  $i^{th}$  chromosome, and  $F(C_i)$  is the fitness value of the  $i^{th}$  chromosome.

Once the fitness of each chromosome is calculated by Equation (5), then the selection process is performed using either the roulette wheel or the ranking method [14]. Pairing is done after the selection process, then the crossover and mutation operations are applied to each paired chromosome to produce new generations (offsprings). Figure 2 presents the crossover and mutation operations. As seen, the genes of the first two chromosomes (parent chromosome I and parent chromosome II) were cut from the third digit on the left and interchanged. This yielded a new pair of chromosomes (offspring I and offspring II). The original values 185 and 234 became 249 and 170, respectively. The mutation operation was applied onto the fourth digit from the left of each offspring by simply reversing digit 1 to 0 and digit 0 to 1, respectively. After the mutation operation, 249 and 170 became 233 and 186, respectively. More details on GA can be found in other studies [11,14,15].

Parent Chromosome-I	1	0	1		1	1	0	0	1	→	185
Parent Chromosome-II	1	1	1		0	1	0	1	0	→	234
<i>after cross-over</i>											
Offspring-I	1	1	1		1	1	0	0	1	→	249
Offspring-II	1	0	1	0	1	0	1	0	1	→	170
<i>after mutation</i>											
New offspring-I	1	1	1	0	1	0	0	1	1	→	233
New offspring-II	1	0	1	1	1	0	1	0	1	→	186

**Figure 2.** Example for crossover and mutation operations.

### 2.3. Particle Swarm Optimization (PSO)

The PSO was developed by Kennedy and Eberhart [16]. The concept is based on the movement of a flock. The individuals in the flock move in search space while adjusting their position and velocity according to the neighboring individuals [17]. The position of an individual,  $x_i$ , is adjusted by Equation (6) [18]:

$$x_{i+1} = x_i + v_{i+1} \quad (6)$$

where  $x_i$  and  $v_i$  are the particle's position and velocity vectors, respectively. The particles velocity,  $v_i$ , is adjusted by Equation (7) [18]:

$$v_{i+1} = \mu [wv_i + c_1r_1(p_i - x_i) + c_2r_2(p_g - x_i)] \quad (7)$$

where  $p_i$  is the position of the best candidate solution;  $p_g$  is the global best position,  $\mu$  is the constriction coefficient ( $\mu = 1$ ) [19],  $w$  is the inertia weight ( $w = 0.4\text{--}0.9$ ) [20],  $c_1$  and  $c_2$  are the acceleration coefficients ( $c_1 = c_2 = 2$ ), and  $r_1$  and  $r_2$  are random values in  $(0, 1)$  [21]. The inertia weight is updated by Equation (8) at each iteration [18]:

$$w_{iter} = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} iter \quad (8)$$

where  $w_{iter}$  is the inertia weight in each iteration,  $iter_{max}$  is the maximum number of iterations, and  $w_{max}$  and  $w_{min}$  are the maximum and minimum inertia weights, respectively. Details of PSO are given in [16,18].

### 2.4. Ant Colony Optimization (ACO)

The ACO was developed by Dorigo [22] who wondered how the ants find the shortest path between their nest and a food source. It was found that ants are completely blind but communicate through a chemical tracer called pheromone. Each ant leaves a pheromone on its path. Hence, it is likely that ants will choose the path having the most concentrated pheromone. This suggests that if there are long and short paths between the nest and the food source, then the ants will eventually choose the shortest path.

The probability that option  $L_{i(j)}$  path is chosen at cycle  $k$  and iteration  $t$  ( $P_{i(j)}(k, t)$ ) can be computed by Equation (9) [23]:

$$P_{i(j)}(k, t) = \frac{\left[P_{i(j)}(t)\right]^\alpha \left[\mu_{i(j)}(t)\right]^\beta}{\sum_{L_{i(j)}} \left[P_{i(j)}(t)\right]^\alpha \left[\mu_{i(j)}(t)\right]^\beta} \quad (9)$$

where  $P_{i(j)}(t)$  is the pheromone concentration associated with option  $L_{i(j)}$  at iteration  $t$ ;  $\mu_{i(j)} = L_{i(j)} / c_{i(j)}$  is the heuristic factor which favors options having smaller local costs;  $c_{i(j)}$  is the set of costs associated with options  $L_{i(j)}$ .  $\alpha$  and  $\beta$  are the exponents controlling the relative importance of pheromone and local heuristic factor, respectively.

The pheromone trail is updated using Equation (10) [23]:

$$P_{i(j)}(t+1) = \delta P_{i(j)}(t) + \Delta P_{i(j)} \quad (10)$$

where  $\Delta P_{i(j)}$  is the change in pheromone concentration associated with option  $L_{i(j)}$  and  $\delta$  is the pheromone persistence coefficient ( $\delta < 1$ ). Employing  $\delta$  has many advantages, such as the greater exploration of the search space, avoidance of premature convergence and costly solutions. More details of ACO algorithm can be found elsewhere [22,23].

## 2.5. Data and Catchment

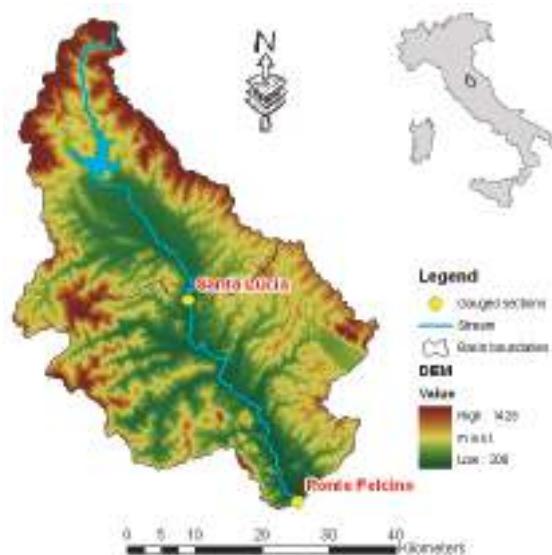
The models were applied to predict real event-based flood hydrographs (Table 1). The hydrographs were recorded along an equipped branch of Tiber River, central Italy (see Figure 3). Santa Lucia station is the upstream station and Ponte Felcino station is the downstream station (Figure 3). Santa Lucia and Ponte Felcino stations have about 935 km<sup>2</sup> and 2035 km<sup>2</sup> drainage areas, respectively. The distance between these two stations is 45 km and the wave travel time of the flood events, as presented in Table 1, is, on the average, almost 3.5 h. The duration of each event is also presented in Table 1.

The ANN model employed the flow stage data recorded at both stations to predict the flow discharge at Ponte Felcino station. Other machine learning models (PSO, ACO; GA), based on the RCM, employed the cross-sectional area data at both stations and flow discharge data at the upstream station to predict the flow discharge at the downstream station. The 4 events, marked as \* in Table 1, were used for calibrating (training in the case of ANN) the models. Since the data was recorded every half an hour and considering the duration of these events in Table 1, the total number of training patterns (input data sets at the calibration stage) amounted to 1012. The total number of data sets used for the validation (testing) stage was 740. More details on the catchment and data can be found in other studies [3,8–10,24].

**Table 1.** Main characteristics of observed flood events.

	Santa Lucia Station		Ponte Felcino Station		$T_L$ (h)	Duration (h)
	$Q_b$ (m <sup>3</sup> /s)	$Q_p$ (m <sup>3</sup> /s)	$Q_b$ (m <sup>3</sup> /s)	$Q_p$ (m <sup>3</sup> /s)		
December 1990	8	418	9	404	2	98
January 1994	35	108	50	241	3	122
May 1995 *	4	71	8	138	4	217
January 1997 *	18	120	36	225	3	77
June 1997 *	5	345	10	449	5	114
January 2003	24	58	50	218	3	150
February 2004 *	22	91	55	276	3	98

$Q_b$  = base flow rate;  $Q_p$  = peak discharge;  $T_L$  = travel time; \* events employed for the training.



**Figure 3.** Tiber River Basin at Ponte Felcino gage site.

### 3. Results and Discussion

#### 3.1. Real Hydrograph Predictions

Flow stage data measured every half an hour at Santa Lucia (upstream station) and Ponte Felcino (downstream station) constituted the input vector of the network, while the flow discharge measured at Ponte Felcino station was the target output. Thus, the network contained 2, 7, and 1 neurons in the input, hidden, and output layers, respectively. The ANN model was trained with 4 events, marked as \* in Table 1 and the total number of training patterns was 1012. The ANN model was trained with  $\delta = 0.04$ ,  $\alpha = 0.02$  and 3000 iterations. The network, by the back propagation algorithm, found the optimal values of the connection weights by minimizing the following objective function:

$$E = \sum_{i=1}^N (Q_d^{obs} - Q_d^{model})^2 \quad (11)$$

where  $N$  is the number of observations,  $Q_d^{obs}$  is the observed discharge at the downstream station, and  $Q_d^{model}$  is the model-predicted downstream station discharge.

The trained ANN model was then employed to predict hydrographs of other three events (December 1990, January 1994, and January 2003 in Table 1). Considering the duration of these events (see Table 1), the total of data sets used in the testing stage was 740. Figure 4 shows the predicted hydrographs, where the performance of ANN was tested against that of the rating curve method (RCM) developed by Moramarco et al. [24] who proposed the relation given by Equation (12):

$$Q_d(t) = \alpha \frac{A_d(t)}{A_u(t - T_l)} Q_u(t - T_l) + \beta \quad (12)$$

where  $Q_u$  is the upstream discharge;  $Q_d$  is the downstream discharge;  $A_u$  is the upstream cross-sectional flow area;  $A_d$  is the downstream cross-sectional flow area;  $T_l$  is the wave travel time; and  $\alpha$  and  $\beta$  are the model parameters, estimated using the same relation, given by Equation (12), for the base flow and peak flow cases [3].

The other methods (GA, PSO, and ACO) were also tested against the RCM. These optimization methods, using the same flow hydrographs (the ones used in the training of the ANN) and by minimizing the same error function (Equation (11)) found the optimal values of the RCM parameters. We herein named them accordingly. For example, when the optimal values of the RCM are found using the GA method, we called it GA\_RCM model. In the GA\_RCM modeling, 80 chromosomes, 80% cross-over rate, and 4% mutation rate were used. The search space was  $(-3, 3)$  and  $(-10, 10)$  for  $\alpha$  and  $\beta$ , respectively. It took 4200, 4600 and 6000 iterations for ACO\_RCM, GA\_RCM and SPO\_RCM, respectively, to reach the optimum solutions. Table 2 presents the optimum values obtained by these methods.

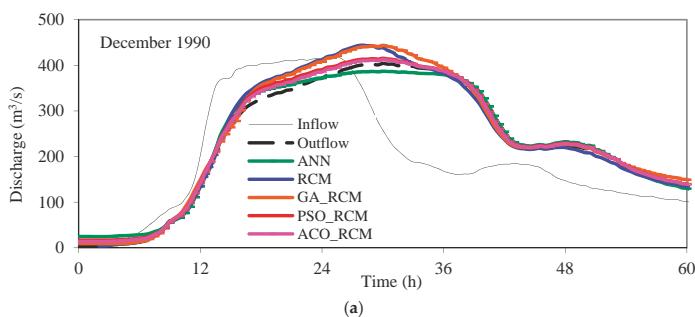
**Table 2.** Optimal values of the Rating Curve Method parameters.

Algorithm	$\alpha$	$\beta$
GA	1.22	-5.86
PSO	1.20	-5.90
ACO	1.23	-5.84

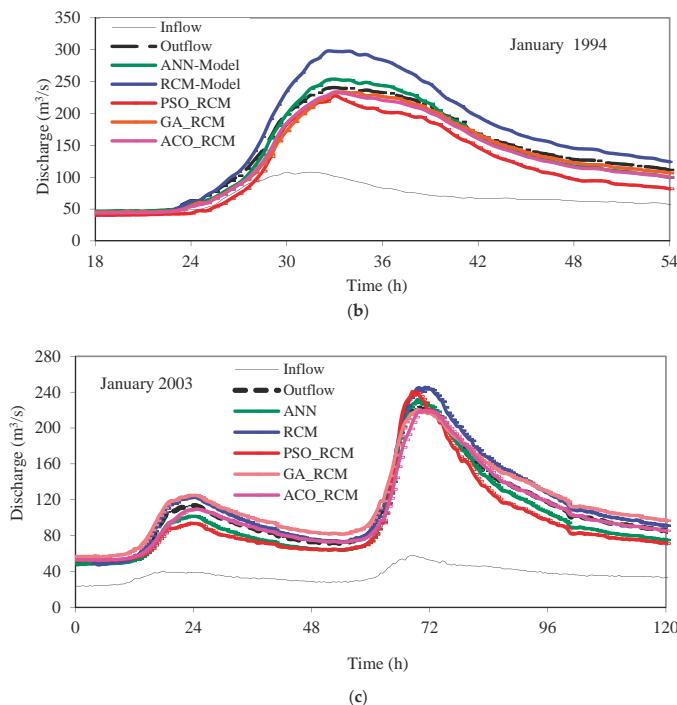
As seen in Figure 4, the machine learning models simulated observed hydrographs satisfactorily. They captured the rising and recession limbs, and peak rates with no delay. Table 3 summarizes the percentage errors for predicting the peak rates and time to peaks. The negative sign means underestimation for the peak rate and early capturing of the peak rate for the timing. As seen, the models made around 5% (or less) error in predicting the peak rates. This error was around 10% for the classical RCM model. The machine learning methods captured the timing of the peaks with less than 1% error, while it was around 5% for the RCM. Also, the mean absolute error (MAE) and the root mean square error (RMSE) values for each run in Figure 4 were presented in Table 3. As seen, the ANN, the GA\_RCM and the ACO\_RCM models produced, on average, around  $7 \text{ m}^3/\text{s}$  MAE and  $9 \text{ m}^3/\text{s}$  RMSE values. These values are, on average,  $\text{MAE} = 11 \text{ m}^3/\text{s}$  and  $\text{RMSE} = 13 \text{ m}^3/\text{s}$  for the PSO\_RCM model. The RCM made high errors as  $13 \text{ m}^3/\text{s}$  MAE and  $17 \text{ m}^3/\text{s}$  RMSE.

Note that the classical RCM needs to estimate different  $\alpha$  and  $\beta$  parameters for each event for the same river reach [8]. On the contrary, herein only a single set of parameters was calibrated by the machine learning algorithms for the same river reach having different inflow flood hydrographs.

Tayfur et al. [8] proposed that ANNs are good interpolators but they cannot be used for the extrapolation purposes. That is, when they are trained by low peak hydrographs they cannot predict high peak ones. On the other hand, Tayfur et al. [10] showed that GA\_RCM does not have this shortcoming of the ANN.



**Figure 4. Cont.**



**Figure 4.** Flood Hydrograph simulations: (a) December 1990; (b) January 1994; and (c) January 2003.

**Table 3.** Error measures ( $E_{Qp}$ : % error in peak discharge, and  $E_{Tp}$ : % error in time to peak).

	December 1990		January 1994	January 2003
	$E_{Qp}$ (%)			
ANN	−5	4	5	
GA_RCM	10	−3	−1	
PSO_RCM	2	−4	10	
ACO_RCM	2	−3	−2	
RCM	10	10	12	
	$E_{Tp}$ (%)			
ANN	0	0	0	
GA_RCM	0	2	0	
PSO_RCM	0	2	−2	
ACO_RCM	0	2	0	
RCM	−10	−2	4	
	MAE ( $m^3/s$ )		Mean	
ANN	8.5	5.6	8.7	7.6
GA_RCM	12.7	4.4	4.7	7.3
PSO_RCM	6.2	12.0	15.4	11.2
ACO_RCM	4.6	3.5	9.1	5.7
RCM	10.4	13.2	14.9	12.8
	RMSE ( $m^3/s$ )		Mean	
ANN	10.3	7.0	9.2	8.8
GA_RCM	15.7	7.1	6.1	9.6
PSO_RCM	8.5	14.7	17.6	13.6
ACO_RCM	6.4	6.3	9.8	7.5
RCM	16.2	17.7	15.9	16.6

### 3.2. Hydrograph Predictions in an Artificial Channel Reach

For this purpose, we considered flood routing in an artificial channel reach of 38 km, with a rectangular cross-section of 40 m width and the Manning roughness value of 0.032. The GA, PSO and ACO methods were used to find the optimal values of the parameters ( $K, x, m$ ) of the nonlinear Muskingum (NMM) flood routing method, which can be expressed as follows [25]:

$$I(t) - O(t) = \frac{dS(t)}{dt} \quad (13)$$

$$S(t) = K[xI(t) + (1-x)O(t)]^m \quad (14)$$

where  $I(t)$ ,  $O(t)$  and  $S(t)$ , respectively, denote the inflow, outflow and reach storage at time  $t$ ; and  $x$  and  $K$ , respectively, denote the parameters of the method known as the weighing parameter and flood wave travel time between inlet and outlet sections of the routing reach, and  $m$  is the exponent of the weighted discharge. We herein name the models GA\_NMM, PSO\_NMM, and ACO\_NMM. For example, if the parameters of the NMM are found by applying the GA method, then it is called herein as GA\_NMM model.

The performance of the ANN, GA\_NMM, ACO\_NMM and PSO\_NMM methods were tested against that of the numerical solutions of the physically-based equations of one-dimensional St. Venant equations:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_l \quad (15)$$

$$\frac{\partial Q}{\partial t} + \frac{\partial (Qu)}{\partial x} + gA \frac{\partial h}{\partial x} = gA(S_o - S_f) \quad (16)$$

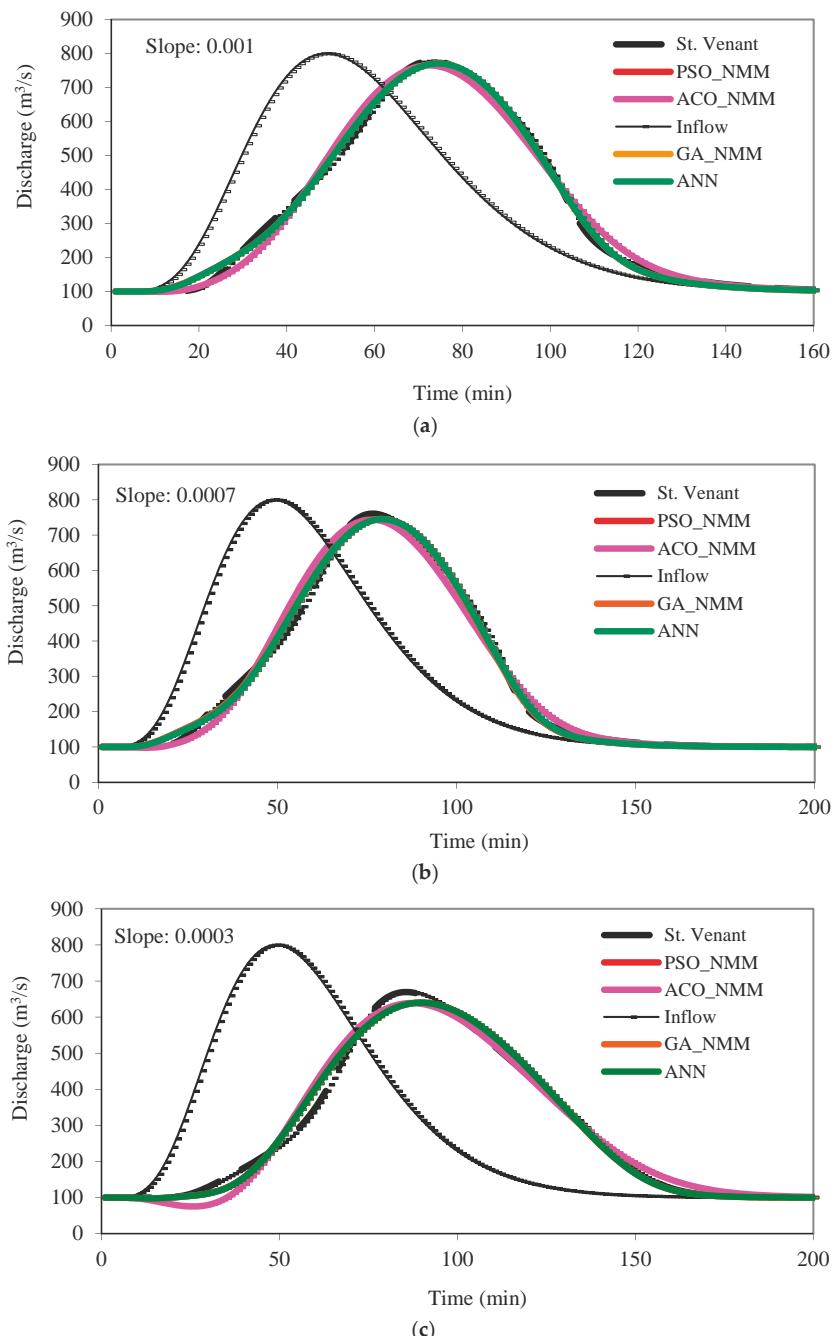
where  $A$  is the cross-sectional flow area,  $Q$  is the flow rate,  $q_l$  is the unit lateral flow,  $g$  is the gravitational acceleration,  $u$  is the flow velocity,  $S_o$  is the channel bed slope and  $S_f$  is friction slope (energy gradient). These equations are well established in the literature [1,2].

By the GA\_NMM, ACO\_NMM and PSO\_NMM, the optimal values of the coefficients ( $K, x$ , and  $m$ ) of the NMM model (see Equations (13) and (14)) were found by minimizing MAE function:

$$MAE = \frac{1}{N} \sum_{i=1}^N |Q_{NMM} - Q_{SV}| \quad (17)$$

where  $Q_{NMM}$  is the computed flow discharge by the NMM (Equations (13) and (14)), and  $Q_{SV}$  is the computed flow discharge by the St. Venant (SV) equations (Equations (15) and (16)).

The inflow hydrograph, similar to the one shown in Figure 5, with maximum peak of 900 m<sup>3</sup>/s, time to peak of 26 h, and base flow rate of 120 m<sup>3</sup>/s, was routed in the same channel reach having three different bed slopes ( $S_o$ : 0.001, 0.0007, and 0.0003) by the St. Venant equations and the generated outflow hydrographs were used to obtain the optimal values of the NMM by the machine learning methods. The search space was 0–1 for  $K$ , 0–0.5 for  $x$  and 1–3 for  $m$ . It took 3000, 3600 and 4000 iterations for GA, ACO and PSO, respectively, to reach the optimum solutions. Table 4 presents the optimal values of NMM. Figure 5 presents the validation runs for a different inflow hydrograph having 100 m<sup>3</sup>/s base flow rate, 800 m<sup>3</sup>/s peak discharge, and 50 h for the time to peak. The inflow hydrograph was routed three times in the same channel reach having different bed slopes. Note that a single set of parameter values given in Table 4 were obtained for the same reach having different bed slopes. Also, note that with the same hydrographs that are used for the calibration of the other methods (GA\_NMM, PSO\_NMM, and ACO\_NMM). In the training stage of the ANN, 5000 iterations and 0.04 for  $\delta$  and  $\alpha$  were employed.



**Figure 5.** Hydrograph simulations: (a) Slope = 0.001, (b) Slope = 0.0007, and (c) Slope = 0.0003.

**Table 4.** Optimal values of the Nonlinear Muskingum Model parameters.

Algorithm	<i>K</i>	<i>x</i>	<i>m</i>
GA	0.0057	0.45	2.20
PSO	0.0056	0.45	2.21
ACO	0.0059	0.45	2.21

As seen in Figure 5, the machine learning algorithms can successfully be employed for simulating flood hydrographs. They are able to capture the peak discharge values as well as the timing of the peaks and the flood volumes, and the rising and recession limbs of the hydrographs. Their performance is as good as that of the St. Venant model. When Perumal et al. [12] investigated the performance of some of the machine learning algorithms against that of the VPMM using a compound channel reach of 40 km, they calibrated the NMM model parameters separately for each river reach having different bed slopes. Herein, a single set of parameters was obtained for the same river reach having different bed slopes.

Any nonlinear search method, such as the multivariate Newton's method, could be used to obtain the optimal values of the NMM. However, the machine learning methods are basically a nonlinear search and optimization methods that do not rely on the mathematical properties of a function, such as the continuity and the differentiability. Hence, they can be used in solving nonlinear, nonconvex, and multimodal problems for which deterministic search techniques incur difficulty or fail completely. Furthermore, the machine learning methods have advantages such as the simple coding, low computational cost, fast convergence, and adaptiveness to new data. The machine learning methods including the recurrent neural networks and neuro-fuzzy networks can also be employed for multi-step-ahead flood and water level forecasting purposes [26,27].

#### 4. Concluding Remarks

The following conclusions are drawn from this study:

1. Machine learning methods can make good predictions of flood hydrographs, using substantially less data, such as easily measurable flow stage. Hence, they can be conveniently adopted for predictions in poorly gauged stations, which is the common case in developing countries. The machine learning methods can be employed in conjunction with the physically based models employing the data acquired by newly developed technologies (the remote sensing, satellite).
2. It is proved by using field data that machine learning algorithms, such as GA, ACO, and PSO are optimization methods without being considered black box models. Since there is a mathematical relation, they both have interpolation and extrapolation capabilities. One more advantage of these models is that one can propose a new equation, such as RCM, provided that it physically makes sense, and by one of these methods, one can find optimal values of the coefficients and exponents of the equation. These methods are robust and efficient and have low computational cost and fast convergence.
3. It is shown that RCM model, whose parameters were optimized by the machine learning algorithm, (GA-RCM, PSO\_RCM and ACO\_RCM), was able to successfully predict event-based individual storm hydrographs having a different magnitude of lateral inflows at the investigated river reach of the Upper Tiber River basin in central Italy. It closely captured the trends, time to peak, and peak rates of the storms with on average, less than 1% and 5% errors, respectively.
4. Likewise, the machine learning-based nonlinear Muskingum models (NMM) can successfully be employed for predicting flood hydrographs. They are able to capture the peak discharge values as well as the timing of the peaks and the flood volumes, and the rising and recession limbs of the hydrographs. Their performance is as good as the St. Venant model.
5. The use of machine learning for discharge prediction is essential for hydrological practices, considering that often, for many river gauging sites, the maintenance is missing, and streamflow

measurements are more and more limited to few strategic gauged river sections. The option to monitor only water levels at gage sites makes these approaches very appealing for their capability to relate, by RCM, local stages and remote discharge.

**Author Contributions:** G.T. and V.P.S. initiated the research. T.M. and S.B. provided the field data. G.T. carried out the simulations and wrote the first rough draft of the manuscript. V.P.S., T.M. and S.B. analyzed the results, made feedbacks and revised the paper.

**Funding:** This research received no external funding.

**Acknowledgments:** The authors wish to thank the Department of Environment, Planning, and Infrastructure of Umbria Region for providing Tiber River data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Henderson, F.M. *Open Channel Flow*; MacMillan: New York, NY, USA, 1966.
2. Chaudhry, M.H. *Open-Channel Flow*; Prentice Hall: Upper Saddle River, NJ, USA, 1993.
3. Barbetta, S.; Franchini, M.; Melone, F.; Moramarco, T. Enhancement and comprehensive evaluation of the Rating Curve Model for different river sites. *J. Hydrol.* **2012**, *464–465*, 376–387. [[CrossRef](#)]
4. Kundzewicz, Z.W.; Napiorkowski, J.J. Nonlinear models of dynamic hydrology. *Hydrol. Sci. J.* **1986**, *312*, 163–185. [[CrossRef](#)]
5. Barbetta, S.; Moramarco, T.; Perumal, M. A Muskingum-based methodology for river discharge estimation and rating curve development under significant lateral inflow conditions. *J. Hydrol.* **2017**, *554*, 216–232. [[CrossRef](#)]
6. ASCE Task Committee. Artificial neural networks in hydrology. I: Preliminary concepts. *J. Hydrol. Eng.* **2000**, *5*, 115–123.
7. Tayfur, G. Modern optimization methods in water resources planning, engineering and management. *Water Resour. Manag.* **2017**, *31*, 3205–3233. [[CrossRef](#)]
8. Tayfur, G.; Moramarco, T.; Singh, V.P. Predicting and forecasting flow discharge at sites receiving significant lateral inflow. *Hydrol. Process.* **2007**, *21*, 1848–1859. [[CrossRef](#)]
9. Tayfur, G.; Moramarco, T. Predicting hourly-based flow discharge hydrographs from level data using genetic algorithms. *J. Hydrol.* **2008**, *352*, 77–93. [[CrossRef](#)]
10. Tayfur, G.; Barbetta, S.; Moramarco, T. Genetic algorithm-based discharge estimation at sites receiving lateral inflows. *J. Hydrol. Eng.* **2009**, *14*, 463–474. [[CrossRef](#)]
11. Tayfur, G. *Soft Computing in Water Resources Engineering: Artificial Neural Networks, Fuzzy Logic, and Genetic Algorithm*; WIT Press: Southampton, UK, 2012.
12. Perumal, M.; Tayfur, G.; Rao, C.M.; Gurarslan, G. Evaluation of a physically based quasi-linear and a conceptually based nonlinear Muskingum methods. *J. Hydrol.* **2017**, *546*, 437–449. [[CrossRef](#)]
13. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
14. Sen, Z. *Genetic Algorithm and Optimization Methods*; Su Vakfi Yayınları: Istanbul, Turkey, 2004. (In Turkish)
15. Goldberg, D.E. Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning. Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 1983.
16. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948. [[CrossRef](#)]
17. Kumar, D.N.; Reddy, M.J. Multipurpose reservoir operation using particle swarm optimization. *J. Water Resour. Plan. Manag.* **2007**, *133*, 192–201. [[CrossRef](#)]
18. Shourian, M.; Mousavi, S.J.; Tahershamsi, A. Basin-wide water resources planning by integrating PSO algorithm and MODSIM. *Water Resour. Manag.* **2008**, *22*, 1347–1366. [[CrossRef](#)]
19. Ostadrahimi, L.; Marino, M.A.; Afshar, A. Multi-reservoir operation rules: Multi-swarm PSO-based optimization approach. *Water Resour. Manag.* **2012**, *26*, 407–427. [[CrossRef](#)]
20. Moghaddam, A.; Behmanesh, J.; Farsijani, A. Parameters estimation for the new four-parameter nonlinear Muskingum model using the particle swarm optimization. *Water Resour. Manag.* **2016**, *30*, 2143–2160. [[CrossRef](#)]

21. Afshar, A.; Shojaei, N.; Sagharjooghifarhani, M. Multiobjective calibration of reservoir water quality modeling using multiobjective particle swarm optimization (MOPSO). *Water Resour. Manag.* **2013**, *27*, 1931–1947. [[CrossRef](#)]
22. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Milano, Italy, 1992. (In Italian)
23. Dorigo, M.; Maniezzo, V.; Colomi, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [[CrossRef](#)] [[PubMed](#)]
24. Moramarco, M.; Barbetta, S.; Melone, F.; Singh, V.P. Relating local stage and remote discharge with significant lateral inflow. *J. Hydrol. Eng.* **2005**, *10*, 58–69. [[CrossRef](#)]
25. Chow, V.T.; Maidment, D.R.; Mays, L.W. *Applied Hydrology*; McGraw-Hill: New York, NY, USA, 1988.
26. Chang, F.J.; Chiang, Y.M.; Ho, Y.H. Multi-step-ahead flood forecasts by neuro-fuzzy networks with effective rainfall-runoff patterns. *J. Flood Risk Manag.* **2015**, *8*, 224–236. [[CrossRef](#)]
27. Chang, F.J.; Chen, P.A.; Lu, Y.R.; Huang, E.; Chang, K.Y. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* **2014**, *517*, 836–846. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

## Article

# Flood Routing in River Reaches Using a Three-Parameter Muskingum Model Coupled with an Improved Bat Algorithm

Saeed Farzin <sup>1</sup>, Vijay P. Singh <sup>2</sup>, Hojat Karami <sup>1</sup>, Nazanin Farahani <sup>1</sup>, Mohammad Ehteram <sup>1,\*</sup>, Ozgur Kisi <sup>3</sup>, Mohammed Falah Allawi <sup>4</sup>, Nuruol Syuhadaa Mohd <sup>5</sup> and Ahmed El-Shafie <sup>5</sup>

- <sup>1</sup> Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan 3513119111, Iran; saeed.farzin@semnan.ac.ir (S.F.); hkarami@semnan.ac.ir (H.K.); N.farahani@semnan.ac.ir (N.F.)  
<sup>2</sup> Department of Biological and Agricultural Engineering, Zachry Department of Civil Engineering, Texas A&M University, 321 Scoates Hall, 2117 TAMU, College Station, TX 77843-2117, USA; vsingh@tamu.edu  
<sup>3</sup> Faculty of Natural Sciences and Engineering, Ilia State University, Tbilisi 0162, Georgia; ozgur.kisi@iliauni.edu.ge  
<sup>4</sup> Civil and Structural Engineering Department, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia; mohmmd.falah@gmail.com  
<sup>5</sup> Civil Engineering Department, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia; n\_syuhadaa@um.edu.my (N.S.M.); elshafie@um.edu.my (A.E-S.)  
\* Correspondence: mohammdehteram@semnan.ac.ir; Tel.: +98-91-6228-1744

Received: 1 August 2018; Accepted: 17 August 2018; Published: 24 August 2018

**Abstract:** Design of hydraulic structures, flood warning systems, evacuation measures, and traffic management require river flood routing. A common hydrologic method of flood routing is the Muskingum method. The present study attempted to develop a three-parameter Muskingum model considering lateral flow for flood routing, coupling with a new optimization algorithm namely, Improved Bat Algorithm (IBA). The major function of the IBA is to optimize the estimated value of the three-parameters associated with the Muskingum model. The IBA acts based on the chaos search tool, which mainly enhances the uniformity and ergodicity of the population. In addition, the current research, unlike the other existing models which consider flood routing, is based on dividing one reach to a few intervals to increase the accuracy of flood routing models. Three case studies with lateral flow were considered for this study, including the Wilson flood, Karahan flood, and Myanmar flood. Seven performance indexes were examined to evaluate the performance of the proposed Muskingum model integrated with IBA, with other models that were also based on the Muskingum Model with three-parameters but utilized different optimization algorithms. The results for the Wilson flood showed that the proposed model could reduce the Sum of Squared Deviations (SSD) value by 89%, 51%, 93%, 69%, and 88%, compared to the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, Pattern Search (PS) algorithm, Harmony Search (HS) algorithm, and Honey Bee Mating Optimization (HBMO), respectively. In addition, increasing the number of intervals for flood routing significantly improved the accuracy of the results. The results indicated that the Sum of Absolute Deviations (SAD) using IBA for the Karahan flood was 117, which had reduced by 83%, 88%, 94%, and 12%, compared to the PSO, GA, HS, and BA, respectively. Furthermore, the achieved results for the Myanmar flood showed that SSD for IBA relative to GA, BA, and PSO was reduced by 32%, 11%, and 42%, respectively. In conclusion, the proposed Muskingum Model integrated with IBA considering the existence of lateral flow, outperformed the existing applied simple Muskingum models in previous studies. In addition, the more the number of intervals used in the model, the better the accuracy of flood routing prediction achieved.

**Keywords:** flood routing; Muskingum model; hydrologic models; improved bat algorithm; Wilson flood; Karahan flood

## 1. Introduction

Flood routing is fundamental to the design of structural, as well as nonstructural, flood control measures [1]. Routing involves the calculation of changes in the magnitude, velocity, and shape of a flood wave, as a function of time at one or several points of the river [2]. There are two types of flood routing methods: Hydraulic and hydrological. Hydraulic methods have complex computations and are more data-intensive, but describe the complete flood wave profile, whereas hydrological methods are much simpler, but yield the flood hydrograph at the end of a reach [3–7]. The hydrologic methods need only the inflow hydrograph for a river reach. A common hydrologic method is the Muskingum method, which has several versions with parameters ranging from two to five. The two and three parameter versions of the Muskingum methods are more popular. In recent years, optimization methods, especially evolutionary algorithms, for estimating the Muskingum parameters have been popular [3]. A brief background of such algorithms is now given.

### 1.1. Background

The flood routing models mainly include two different types of modeling: the hydraulic and hydrologic models. The hydraulic model is usually developed in a one or two-dimensional domain. Full three water shallow models and two diffusive models were used for an urban site, and the results had the same difference with each other because of different representation of a numerical and hydraulic method in the model algorithm process [5]. Hunter et al. [6] successfully set three explicit hydraulic models based on the inertia, diffusive, and shallow water models for flood simulation. The results indicated that the models with the shallow water equation were simple and could provide good accuracy, for the prediction of depth and velocity of the flood [5]. Dottori and Todini [7] evaluated two-dimensional models based on the diffusive wave for urban floods, and the results indicated that the model could simulate the overall phenomenon well. Kim et al. [8] evaluated the different meshes in diffusive models, to investigate the effect of different meshes on the flood hydrograph. Prestininzi [9] applied the diffusive models based on the impulsive wave for inundation areas, and the results showed the model could simulate the flood conditions even in complex topography, based on a good match of simulated results with the observed data. Aricò et al. [10] applied diffusive wave equation based on 2-D numerical models for a slow varying flood, and the results showed that the simulated depth of the flood had a relatively good match with observed data. Classical, explicit finite differences in the hydraulic models and simple Muskingum model were used to investigate the flood routing [11]. It has been reported that the applied numerical methods had numerical instability, for some case studies. As a result, the Muskingum model showed superior performance compared to the same applied hydraulic models [11]. It has been reported that the 2-D models could be developed, based on the availability of enough information about topography and topology. With this information, a 2-D hydraulic model could successfully simulate the flood characteristics for different urban conditions. In fact, digital maps helped to identify all the required information about the boundary conditions, and to differentiate between numerous transitions within the urban hydraulic modes [12].

In addition, it has been reported that the 2-D models had trouble in application to particular cases under small water depths, especially when the status comes close to wet/dry boundary conditions, so that there is a need for specific algorithms for simulation [6]. Costabile et al. [13] reported that the main advantages of the one-dimensional model over the 2-D models for flood routing, are a simpler run process and low computational time where topographic data was unnecessary.

Fassoni-Andrade et al. [14] considered the development of a one-dimensional model based on the equation of hydrological models, which include the continuity equation and mass equation, such as the equation of the Muskingum models [14]. It was observed that one-dimensional flow routing inertial models, based on the explicit solution were superior to the other models. These models simplified

the Saint-Venant equation, and the main advantages of these models were good simulated results with a simple structure. Singh and Arvamuthan et al. [15] applied two hydraulic models that were developed based on the Kinematic and diffusion waves, in addition, the results were compared to the Muskingum model for the flood routing. The results showed that the simulation of hydraulic models was dependent on the kinematic wave number, so that when the value of this parameter was not considered based on accurate computation, the results for the hydraulic model could be worse than hydrologic models. Costabile et al. [16] reported that 2-D models could overcome the limitation of 1-D models, when the case study characterized as unsteady flow in irregular topography. The reports showed that if the flow was not one dimensional for the urban hydraulic, the one-dimensional channel network should be used instead of a one-dimensional model. In addition, the results showed the significant difference between 1-D and 2-D models to simulate the velocity and depths.

However, the results showed that the complex nonlinear form characteristic, numerical stability, high computational time, and complexity in the run process of hydraulic models, meant that the simpler and more accurate models have high importance [13]. In fact, the hydraulic models need to measure the flow depth and discharge based on applying stream gaging. These models are known as complex models and difficult to use, whilst the hydrologic models need only to use the discharge data. In addition, the hydrologic models can be effective for the initial planning level, where the measuring system is undeveloped for accurate measurement [13]. For example, Chatila [17] simulated flood routing based on the Muskingum model and EXTRAN hydraulic model. The hydraulic model developed was based on finite difference. Both hydrologic models and hydraulic models, were applied on simple and compound channels for flood routing. The results revealed that the Muskingum model had achieved higher accuracy compared with the hydraulic model because of its flexibility in calibration, where even the river bed geometry was not considered for this model. It has been demonstrated that the Muskingum model could simulate the peak discharge, achieving a close fit with the actual one, compared to the hydraulic model. Furthermore, it has been reported that hydraulic models are dependent too many assumptions, such as reach geometry, channel slope, and flow velocity, which causes the application of some hydraulic models to be limited to the specific case studies.

The Muskingum model is a useful and important hydrological model, due to its high accuracy and simplicity. Hydrological models could be accomplished after estimating the value of parameters, on the other hand, hydraulic models are required to simulate the complex boundary hydraulic conditions that causes an increase in the computational time [17].

Therefore, this model was used as a model with free access, fast computation, highly accuracy, and low cost. Furthermore, it can be used as a good tool, instead of complex hydraulic models, for flood simulation. Additional background of the application of the Muskingum model and its integration with an evolutionary algorithm, will be presented and discussed hereinafter [18].

Under a two parameter Muskingum method, Luo and Xie [19] applied the immune clonal selection algorithm (ICSA) for flood routing in a river in China, and found that the algorithm had faster convergence than the GA and PSO; and routed discharges had a high correlation with observed discharges.

Geem [20] obtained the two parameters of Muskingum method using a harmony search algorithm (HAS) for the Wilson flood in the USA, and obtained less root mean square between the predicted and observed discharges, than for GA and PSO, and less computational time.

Nelder-Mead simplex algorithm (NMSA) was considered for flood routing, and a case study in the USA [21]. The parameters of the Muskingum model were considered as decision variables, and the results indicated that the RMSE (root mean square error) based on NMSA decreased by 20%, compared to the genetic algorithm [21].

Karahan et al. [22] applied a hybrid of GA, HAS, and nonlinear programming to a three-parameter Muskingum method for flood routing in a river, and found the hybrid algorithm more accurate.

Orouji et al. [23] used a genetic programming algorithm (GPA) for flood routing by the Muskingum method, and showed that GPA was more accurate than GA and PSO. The Muskingum model by 4 parameters was considered under different case studies [23].

Four parameters were considered as decision variables, and the results indicated that the model based on the considered parameters with the genetic algorithm decreased the RMSE and mean absolute error (MAE) by 20% and 25%, respectively, compared to the nonlinear programming methods [24].

The hybrid PSO and harmony search algorithm was considered for flood routing [25]. The results indicated that the new hybrid method could increase the convergence velocity of the harmony algorithm, and decreased the error indexes, RMSE, and MAE, compared to the simple harmony and particle swarm algorithm.

Ouyang et al. [26] applied a hybrid of PSO and GA for Muskingum flood routing and showed that the hybrid algorithm was faster, and more accurately predicted the peak discharge and time to peak.

Under the three-parameter Muskingum method, Geem [27] found the harmony algorithm (HA) to have higher convergence than PSO and GA. Under the four-parameter Muskingum method, the Frog Leaping Algorithm (FLA) was found to have a lower computational error, than PSO and GA [28]. Niknazar and Afzali [29] used an improved Honey Bee Algorithm (IHBA) to optimize three parameters of the Muskingum method, and found it to be superior to GA and PSO.

Using an Invasive Weed Optimization Algorithm (IWOA) for parameter optimization, Hamedи et al. [2] found the five-parameter Muskingum method to be more accurate than the four-parameter version. With PSO for parameter estimation, Moghadam et al. [30] found the four-parameter Muskingum method to be more accurate, than the three-parameter Muskingum method with GA and linear programming.

Using the gravitational search algorithm for parameter optimization, Kang et al. [31] found the four-parameter Muskingum method to be accurate for flood routing. Flood routing for a case study in China based on real code genetic algorithm was considered [3], and the results indicated that the four-parameter Muskingum model based on genetic algorithm decreased the RMSE and MAE of the two- and three-parameter Muskingum models.

Barati et al. [21] applied different kinds of GPA to the four-parameter Muskingum method for flood routing, and found the fixed genetic programming to be more accurate. For flood routing using the Muskingum-Cunge method, Wang et al. [23] found PSO to yield better results than GA.

In 2018, Lee [32] developed and applied an advanced Muskingum flood routing model by considering continuous stream flow, utilizing weighted inflow. Several statistical indicators have been used to evaluate the performance of the suggested model. The proposed model provided acceptable results, compared to those obtained from previous studies. The results showed that the vision corrected algorithm (VCA) had experienced a relatively small error index compared to the GA, PSO, and nonlinear programming models, for different flood case studies.

The literature review showed that the evolutionary algorithm has a high ability for obtaining the parameter values of the Muskingum model, but some algorithms have limitations, such as trapping in local optimums, slow convergence velocity, or insufficient accuracy for the simulation [33]. For example, the GA can trap in the local optimums or the PSO may have an immature solution due to fast convergence [1,27]. Some evolutionary algorithms have many random parameters, and accurate determination of these parameters is difficult. Thus, the improvement of previous algorithms or definition of new algorithms is necessary.

## 1.2. Problem Statement

Flood routing is nonlinear and multimodal, with noise. The most widely used model for flood routing is the Muskingum model. One of the main components of the Muskingum model procedure, is the inclusion of a particular number of parameters ranging between two and five, which have to be estimated based on the case study characteristics to be able to accurately route the flood. There are different procedures to estimate the Muskingum model parameters. Actually, the better

the estimation of the parameters' value, the better the prediction of the flood routing characteristics achieved [33–37]. Having an optimization algorithm that is able to optimize the value of these parameters, whether two or three, is needed to enhance the ability of the Muskingum model to identify the flood routing characteristics. Thus, a robust algorithm that leads to the global optimal solution is needed. Bats exhibit a mysterious behavior that has long been attractive. They are able to orientate themselves to their surroundings and food acquisition, without depending on their eyesight. Bats consistently emit echolocation signals. Through analyzing the returning echoes in the auditory system, bats can distinguish their environment and find preys. By continuously watching and concentrating on the abilities of bats, scientists have suggested different bat-inspired algorithms (i.e., bat algorithm (BA) and bat intelligence (BI) algorithm) for the solution of optimization problems [33,34]. Bozorg-Hadad et al. [35] used BA to optimize the use of a repository to reduce hydroelectricity energy shortage. This algorithm has been shown to have a faster convergence, than GA and PSO. Ahmadinafar et al. [36] used a hybrid BA to exploit a 10-repository system to increase energy production, which reduced computational time compared to other evolutionary algorithms. Studies have shown that BA is a powerful method but it has weaknesses, such as trapped in the local optimum or premature convergence [37,38]. Thus, it needs to be improved.

### 1.3. Objective

In the light of the above, the use of Muskingum model showed its success when applied in flood routing prediction, but it has a few limitations. For example, it cannot be used for the complex boundary condition. In addition, one of the weaknesses of the model is the consideration of the lateral flow. In fact, most of the previous researches ignore the lateral flow while using the simple Muskingum model. Therefore, if lateral flow exists and has high volume, the Muskingum model simulates the flood without consideration of lateral flow [23–31], and hence, the achieved accuracy is relatively low. In fact, simplification of the Muskingum equation was a reason to omit the effect of lateral flow for flood routing, whilst there is lateral flow in the reach when the flood happens in nature. Although, there are a few number of references considering the lateral flow, some of them are limited to specific case studies with the low flow lateral condition and using Muskingum models with a simple structure [39–42].

Limited studies showed that the consideration of the lateral flow, with the help of hydraulic models, could simulate the actual situations that close to the real conditions [12]. Osolivan et al. [17] reported the disadvantages of the hydraulic models for flood routing in the floodplain. Their reports showed that the initial and boundary conditions, and the resistance characteristics of main channels in the floodplain, are necessary for the hydraulic modeling.

The present study develops the new Muskingum model for flood routing considering lateral flow. In addition, the Muskingum Model has been coupled with a new Improved Bat Algorithm (IBA), to optimize the estimation of the three-parameter Muskingum model.

The objective of this study, was to couple the three-parameter Muskingum method (TPMM) with an improved bat algorithm (IBA) for flood routing, with a multi-reach method and consideration of lateral flow. Citing easy trapping in the extremum for bat algorithm, an improved bat method is suggested and used to simulate flood routing. Chaos search tool is defined to enhance the uniformity and ergodicity of the population. Adapting weight is defined to balance the local and global search tools, for the bat algorithm. One of the advantages of the new bat algorithm is related to decreasing the search range, based on dynamic contraction.

The innovation of the study is the use of a new bat algorithm for flood routing lateral flow. Moreover, whilst previous studies usually considered one reach for flood routing without lateral flow, the present study compared the effect of dividing a river into different reaches on flood routing, and the prediction of peak discharge.

## 2. Materials and Methods

### 2.1. Flood Routing

The Muskingum method of flood routing is based on the continuity equation and a storage-discharge relation [30]. The present study considers the lateral flow for the flood routing for the case studies based on a ration of inflow rate  $O_{lateral} = \beta I_t$ , while the other studies do not consider the effect of lateral flow for flood regime, and thus it adds one term to the equation of the Muskingum model with three parameters. If  $\beta$  coefficient equals to zero, the lateral flow has not been considered for the flood routing.

$$\frac{ds_t}{dt} = O_t - (1 + \beta)I_t \quad (1)$$

$$S_t = K[X(1 + \beta)I_t + (1 - X)O_t] \quad (2)$$

where  $O_t$  is the output flow at time  $t$ ,  $I_t$  is the input flow at time  $t$ ,  $S_t$  is the storage at time  $t$ ,  $\frac{ds_t}{dt}$  is the storage time variation at time  $t$ ,  $K$  is the time coefficient of storage, and  $X$  is the weighting factor showing the effect of input and output flows on storage.

Equation (2) expresses a linear relation between storage, and input and output flows. However, a non-linear has also been presented as:

$$S_t = K[X(1 + \beta)I_t + (1 - X)O_t]^m \quad (3)$$

where  $m$  is an exponent. Using Equations (1) and (3), one can obtain [21]:

$$O_t = \left( \frac{1}{1 - X} \right) \left( \frac{S_t}{K} \right)^{\frac{1}{m}} - \left( \frac{X(1 + \beta)}{1 - X} \right) I_t \quad (4)$$

$$\frac{\Delta S_t}{\Delta t} = - \left( \frac{1}{1 - X} \right) \left( \frac{S_t}{K} \right)^{\frac{1}{m}} + \left( \frac{1(1 + \beta)}{1 - X} \right) I_t \quad (5)$$

Using  $S_t, \Delta S_t$ , the storage later can be expressed as:

$$S_{t+1} = S_t + \Delta S_t \quad (6)$$

Flood routing can be done using the following steps:

1. Consider initial values for parameters  $K, X, \beta$ , and  $m$  and enter them into the optimization algorithm, in the form of initial population.
2. Calculate the storage based on Equation (3), assuming the equality of input and output flow.
3. Calculate the change in storage relative to time, based on Equation (5).
4. Calculate the storage based on  $t + 1$ , according to Equation (6).
5. Calculate the output flow at  $t + 1$ , based on Equation (4).
6. Repeat steps 2 to 5.

### 2.2. Optimization of Multi-Reach Muskingum Coefficients

The multi-reach Muskingum method is introduced to enhance the accuracy of the Muskingum method. The river under study was divided into several smaller reaches, and for each reach, routing was done separately. In other words, for each reach, parameters  $X, K, \beta$ , and  $m$  were calculated separately, and the output hydrograph was obtained based on the input flood hydrograph and the assumed values of  $X, K, \beta$ , and  $m$  for the first reach [3,21,28,31]. This output hydrograph was considered as the input hydrograph for the second reach, and so on. For the second reach, the assumed values of  $X, K, \beta$ , and  $m$  were used, and the output hydrograph was calculated. This process was repeated for all the reaches, until the output hydrograph of the last was obtained. By comparing

the computed hydrograph with the observed hydrograph, the error was calculated, and to reach the minimum error, Muskingum coefficients were optimized at all reaches. Figure 1 shows the division of the river into several reaches.

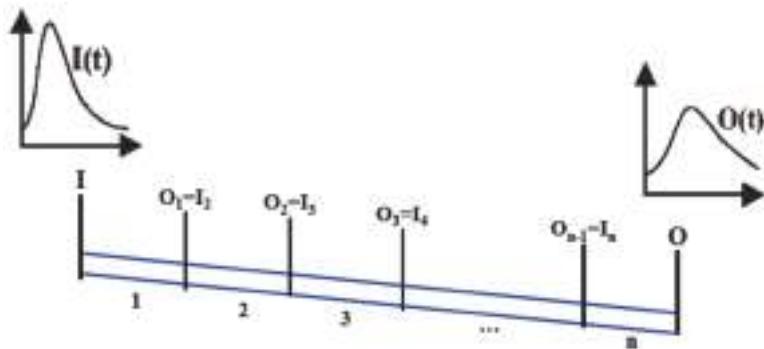


Figure 1. Discretization of the river stream.

### 2.3. BAT Algorithm

Bat algorithm (BA) is based on bat sound reproduction and sound reflection. The difference in loudness that comes from the surrounding environment, allows the bat to identify the barrier from food. Bats produce very high sound pulses and listen to their return from the objects around. Each pulse remains only for a few milliseconds. BA is based on the following assumptions [30,39–41]:

1. All bats have a high ability to receive sound, so that they can detect food after producing loud sounds.
2. Bats fly randomly at a velocity at place  $y_l$ , capable of producing sound with  $f_{\min}$  frequency and  $\lambda$  wavelength. The sound produced by bats also has loudness  $A_0$ .
3. The loudness of sound, of the bats ranges from  $A_0$  to  $A_{\min}$ .

Each sound produced by the bat has a pulse rate ( $r$ ), between 0 and 1. The sound frequency speed and position of the bats are updated as:

$$f_l = f_{\min} + (f_{\max} - f_{\min}) \times \beta \quad (7)$$

$$v_l(t) = [y_l(t-1) - Y_*] \times f_l, t = 1, 2, \dots, T \quad (8)$$

$$y_l(t) = y_l(t-1) + v_l(t), t = 1, 2, \dots, T \quad (9)$$

where  $f_l$  is the frequency of sound of bats,  $f_{\min}$  is the minimum frequency,  $f_{\max}$  is the maximum frequency,  $\beta$  is the random coefficient between 0 and 1,  $v_l(t)$  is the velocity of the bat,  $Y_*$  is the best position of the bat,  $y_l(t)$  is the position of the bat, and  $T$  is the number of periods evaluated.

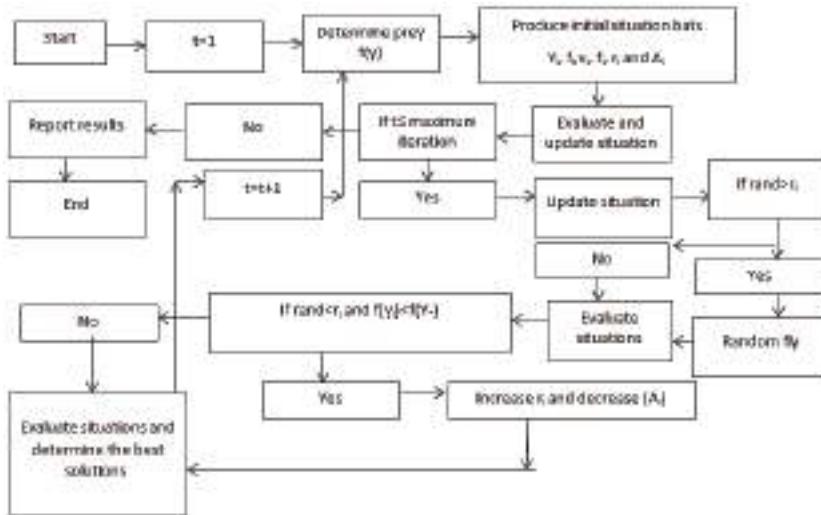
The following equation is used for local search in the bat algorithm:

$$y(t) = y(t-1) + \varepsilon A(t), t = 1, 2, \dots, T \quad (10)$$

where  $\varepsilon$  is the random variable between  $-1$  and  $1$ , and  $A(t)$  is the loudness of sound. Loudness and pulse rates are updated, according to each stage of the iteration. For example, zero sound loudness means that the bat has found its prey and has temporarily stopped the search.

$$r_l^{t+1} = r_l^0 [1 - \exp(-\gamma t)] A_l^{t+1} = \alpha A_l^{t+1} \quad (11)$$

where  $\alpha$  and  $\gamma$  are fixed as constant coefficients. For any value of  $\alpha$  between 0 and 1, and  $\gamma$  greater than zero,  $A_l^t \rightarrow 0$  and  $r_l^t \rightarrow r_l^0$  are true. Figure 2 shows the mathematical procedure of the bat algorithm. In addition, it should be noted that the random walk is considered as a parameter for the local search, for the bat algorithm.



**Figure 2.** The flow chart of the bat algorithm.

#### 2.4. Improved Bat Algorithm (IBA)

The initial arrangement for the initial version of the BA is defined randomly, and it can be a reason for the uneven distribution that causes premature convergence. The chaos is a technique for improving different algorithms, where the basic idea is related to the exchange of members in the range of  $(-1, 1)$ . The logic mapping function is used for modulating the algorithms. Then individuals are inserted into the chaos sequence, so that it should satisfy the chaos variable space. Then, linear transformation is used to return the members to the corresponding position. The convert space is shown based on following mathematical equation:

$$L_i = \frac{2(x_i^a - a)}{(b - a)} - 1 \quad (12)$$

where  $x_i^a$ : the initial position of the members.

The following equation is used to show the logic mapping function:

$$L_{i+1} = 1 - 2 \times L_i^2 \quad (13)$$

Then, the elements and values are returned to the corresponding position by the following linear transformation:

$$y_i^0 = \frac{1}{2(b - a)} L_i + \frac{1}{2(b + a)} \quad (14)$$

Furthermore, adapting weight is applied to the bat algorithm to have a good balance between global search ability and local search ability.

$$y_l^t = w(t) \cdot y_l^{t-1} + v_l^t \quad (15)$$

The weight is computed based on the following equation:

$$w(k) = \frac{w_{\max} \cdot (w_{\max} - w_{\min}) \cdot (T_{\max} - k)}{T_{\max}} \quad (16)$$

$w_{\max}$ : the initial weight,  $w_{\min}$ : the final weight, and  $k$ : the current iteration number.

The final level is related to the application of dynamic contraction, for adjusting the convergence speed:

$$\begin{aligned} y_{\min,i} &= \max\{y_{\min,i}, x^* - rand \times (y_{\max,i} - y_{\min,i})\} \\ y_{\max,i} &= \min\{y_{\max,i}, x^* - rand \times (y_{\max,i} - y_{\min,i})\} \end{aligned} \quad (17)$$

where  $y_{\min,i}$ : the lower bound position, and  $y_{\max,i}$ : the upper bound position.

The algorithm functions using the following steps:

1. Adjust the random parameters for the algorithm, such as loudness, pulsation rate, frequency, and other parameters.
2. The individual position is computed using Equations (13)–(15), and then the objective function is computed for each member, and the best solution is considered as  $Y_*$ .
3. The frequency and velocity are updated using Equations (7) and (8), and the position is computed using Equation (17).
4. The randomness value is compared with  $r_l$ , and if  $r_l$  is less than the randomness value, the distribution of the best position is acted based on 0.01 times the random disturbance.
5. The local search is considered for this level. If the loudness is less than rand, the loudness should be updated and the pulsation rate should be improved using Equation (12).
6. Compute the objective function and change the range using Equation (16).
7. The convergence criterion is checked and if it is satisfied, the algorithm finishes or else the algorithm goes to step 2.

## 2.5. Genetic Algorithm (GA)

In GA, the initial version of the population is composed of different solutions [40]. During an iterative process, subsequent populations are generated to improve the objective function. At each stage, some members from the current population are selected to generate individuals or children of the next generation, based on the fact that the likelihood of selecting people with better performance than others is more likely [41]. The selected individuals produce the next population based on two genetic operators, composition and mutation. The following equations can be used for the composition operator [33].

$$Pop_i^{new} = \alpha Pop_i^{old} + (1 - \alpha) Pop_j^{old} \quad (18)$$

$$Pop_j^{new} = \alpha Pop_j^{old} + (1 - \alpha) Pop_i^{old} \quad (19)$$

where  $Pop_i^{new}$  is the  $i$ -th child,  $Pop_i^{old}$  is the  $i$ -th parent,  $Pop_j^{old}$  is the  $j$ -th parent,  $Pop_j^{new}$  is the  $j$ -th child, and  $\alpha$  is a coefficient between 0 and 1. Moreover, mutation is based on the following equation:

$$Pop_{i,j}^{new} = Var_{i,j}^{low} + \beta (Var_{j,i}^{hi} - Var_{j,i}^{low}) \quad (20)$$

where  $Var_{i,j}^{low}$  is the lower limit of the  $i$ -th gene in the  $j$ -th chromosome,  $Var_{j,i}^{hi}$  is the upper limit of the  $i$ -th gene in the  $j$ -th chromosome, and  $\beta$  is a random coefficient between 0 and 1. In composition, the production of both new individuals is done by changing the gene. The mutation operator is used for the change in chromosomes and transforming their genes to create diversity in the population.

## 2.6. Particle Swarm Algorithm (PSO)

Initially, the process starts with a particle set. Each particle is considered as a random solution. In the next step, searches are performed sequentially to achieve the optimal answer. The  $i$ -th particle is associated with a position in an  $s$ -dimension space, where the value of  $s$  shows decision-making variables of the problem [42]. The values of  $s$  variables, which determine the positions of particles are a possible solution for the optimization problem. Each particle  $i$  is completely determined by three vectors. Vector  $X_i$  is the current position of the particle,  $Y_i$  is the best position where the particle is iterated, and the vector of the particle velocity is shown by  $V_i$ . Then the particle position and particle velocity vector are updated as:

$$V_i^{iter+1} = wV_i^{iter} + c_1rand(Y_i^{iter} - X_i^{iter}) + c_2rand(Y_*^{iter} - X_i^{iter}) \quad (21)$$

$$X_i^{iter+1} = X_i^{iter} + V_i^{iter+1} \quad (22)$$

where  $V_i^{iter+1}$  is the new velocity of the particle, the personal learning coefficient  $c_2$  is the global learning coefficient,  $Y_*^{iter}$  is the best solution among the solutions, and  $X_i^{iter+1}$  is the new position of the particle. Moreover,  $w$  is the coefficient of inertia.

### Indices of Error Measurement

1. The sum of squared deviations (SSD): SSD index is used as the objective function in the present study. The index calculates the total of squared deviations between observed and real discharges [28,42–46]:

$$\text{Minimize}(SSQ) = \sum_{t=1}^n (O_{bt} - O_{st})^2 \quad (23)$$

where  $O_{obt}$  is the observed discharge,  $O_{st}$  is the simulated discharge, and  $n$  is the number of data.

2. The sum of absolute deviations (SAD): SAD is the total sum of total deviations between observed and predicted discharges [20,30]:

$$\text{Minimize}(SAD) = \sum_{i=1}^n |O_{bt} - O_{st}| \quad (24)$$

3. Error of Peak discharge (EP): EP index measures the difference between predicted and observed discharges [43–46].

$$EQ_p = \frac{|O_{observed}^{peak} - O_{routed}^{peak}|}{O_{observed}^{peak}} \quad (25)$$

4. Error of time to peak (ETP): The ETP index measures the difference between predicted and observed time differences of discharge [24,37,38].

$$ET_p = [T_{observed}^{peak} - T_{routed}^{peak}] \quad (26)$$

$T_{observed}^{peak}$  is the observed discharge, and  $T_{routed}^{peak}$  is the time related to the routed discharge.

5. Mean absolute relative error (MARE): The mean of the relative error between observed and predicted discharges:

$$MARE = \frac{1}{N} \sum_{i=1}^n \frac{(Q_t^{observed} - Q_t^{routed})}{Q_t^{observed}} \quad (27)$$

$N$  is the number of data.

6. Varex Q (Variance index): This indicator shows the proximity of predicted and observed hydrographs with each other.

$$VarexQ = \left[ 1 - \frac{\sum_{i=1}^N (O_i^{observed} - O_i^{routed})}{\sum_{i=1}^N (O_i^{observed} - O_{mean}^{observed})} \right] \times 100 \quad (28)$$

$O_{mean}^{observed}$  is the observed average discharge. The closer the coefficient is to one, the more accurate the ability to predict the flood will be.

7. The agreement index ( $d$ ) based on follow equation, shows the performance of the model well, so that the value of index can change from 0 to 1 [45,46].

$$d = 1 - \frac{\sum_{i=1}^N (O_i^{observed} - O_i^{routed})^2}{\sum_{i=1}^N (|O_i^{routed} - \bar{O}_i^{observed}| + |O_i^{observed} - \bar{O}_i^{observed}|)} \quad (29)$$

$\bar{O}_i^{observed}$ : average of observed data.

### 3. Results and Discussion

This paper considered three case studies for flood routing. Two case studies were considered as bench problems, which have been used by different researchers using many methods for flood routing (Wilson and Karahan floods), and one case study was related to a river in Myanmar that had an important flood. The Wilson flood is considered as an important case study and different researchers tested different algorithms on this case study [2,3,28–30], and thus a comprehensive study can be considered for this case study. The Karahan flood is considered as one of the case studies that have been investigated by different researchers as a benchmark problem [28–31,35].

#### 3.1. Wilson Flood

Wilson Flood [44] is one of the most important benchmark applications, to investigate the performance of the Muskingum model and other hydrologic models. In fact, this flood pattern was generated under experimental conditions. Different mathematical models were examined using this flood data pattern, which received great attention from researchers in examining their models.

The data was extracted from Wilson [44]. This information includes single peak inflow and outflow hydrographs with lateral flow. In addition, the applied algorithms in this research consider the lateral flow, although as shown in Table 4, previous researches have ignored the lateral flow because of low value, as shown in Table 6.

Several methods, such as the Segmented Least Square Method (SLSM), Hook and Jeeves (HJ) method, in combination with the Conjugate Gradient (HJ + CG), HJ method, in combination with Davidson Fletcher-Powell (HJ + DFP), nonlinear least squares (NONLER), Genetic Algorithm (GA), Harmony Search (HS), Particle Swarm Optimization (PSO), and Honey Beaming Optimization (HBMO), have used this flood without consideration of lateral flow. Given that evolutionary algorithms have random parameters, sensitivity analysis was used to determine the exact values of parameters. The evolutionary algorithms have random parameters, where the accurate values of these algorithms are computed based on sensitivity analysis. It means that the variation of objective function is determined versus the variation of parameter values, and when the objective function has the best value for a parameter, the value of this parameter is introduced as the optimal. The SSD was considered as an objective function for the current study. The frequency parameters were used to update the

velocity and then the position of bats was computed based on velocity. When the objective function value is minimized, the value of the different parameters is considered at its optimal value.

Tables 1–3 show the sensitivity analysis of IBA parameters, BA, GA, and PSO for flood routing in a single reach. The objective function was considered SSD for this study.

The best population associated with IBA was 60, with the lowest SSD. Moreover, the maximum frequency was 5, with the objective function as 5.01. The maximum loudness of sound was 0.6, with a random walk rate of 5. Furthermore, the mutation rate for GA was 0.6, and the recombination rate for GA was 0.7. In addition, the personal and global learning coefficient in PSO was 2, and the inertia coefficient was 0.6. When the inertia coefficient was 0.6, the objective function had the least value (10.82), and thus, the best value for the inertia coefficient was selected to be equal to 0.6 for the PSO. Other parameters can be seen in the Tables 1–3.

**Table 1.** Analysis for the performance of Improved Bat Algorithm (IBA) algorithm (Wilson flood).

SSD							
Objective Function (cms)	Random Walk Rate	Objective Function (cms)	Maximum Loudness	Objective Function (cms)	Maximum Frequency	Objective Function (cms)	Population Size
6.23	1	6.01	0.2	6.12	1	6.23	20
5.66	3	5.89	0.4	5.78	3	5.89	40
4.12	5	4.12	0.6	4.12	5	4.12	60
5.14	7	5.24	0.80	5.76	7	5.15	80

**Table 2.** Analysis for the performance of the Genetic Algorithm (GA) (Wilson flood).

SSD					
Objective Function	Crossover Rate	Objective Function (cms)	Mutation Rate	Objective Function (cms)	Population Size
46.12	0.10	47.12	0.20	45.39	20
43.21	0.30	42.24	0.40	38.94	40
39.19	0.50	39.24	0.60	39.23	60
40.12	0.70	40.23	0.80	40.12	80

**Table 3.** Analysis for the performance of the Particle Swarm Optimization (PSO) (Wilson Flood).

SSD							
Objective Function (cms)	w	Objective Function (cms)	c2	Objective Function (cms)	c1	Objective Function (cms)	Population Size
12.22	0.2	11.21	1.6	12.11	1.6	12.24	10
10.90	0.4	10.89	1.8	11.89	1.8	10.45	30
10.82	0.6	10.80	2.0	10.82	2.0	10.80	50
11.32	0.8	11.12	2.2	11.24	2.2	11.23	70

Table 4 compares IBA with other evolutionary algorithms, for a single reach routing. The SSD value for IBA was 4.123, with SSD being reduced by 89%, 51%, 93%, 69%, 88%, and 97% compared to GA, PSO, PS, HS, HBMO, and SLSM, respectively. In addition, the SAD index was 7.112, reduced by 69%, 22%, 75%, 69%, 81%, and 84% relative to GA, PSO, PS, HS, HBMO, and SLSM, respectively. In addition, EP and MARE showed the superiority of IBA to other methods. VarexQ index for IBA, compared to other methods, showed more consistency with the predicted hydrograph. Furthermore, the performance of IBA was better than BA, so that SSD, SAD, and other error indexes for IBA were less than BA. For example, SSD and SAD for IBA were 4.123 and 7.112, whilst SSD and SAD for BA were 5.123 and 8.114. In addition, focusing on the peak discharge value, it could be depicted that the computed peak discharge based on IBA (85.11) was the most nearest estimated value to the real observed one (85), and in general, showed a worthy match with observed discharge during the whole period. In addition, for the estimation of the peak time which was predicated based on IBA, it was same to the observed

one 60 h. For further assessment, Table 5 shows the performance of all the models, and it could be observed that the proposed IBA in the present study outperformed all the other models.

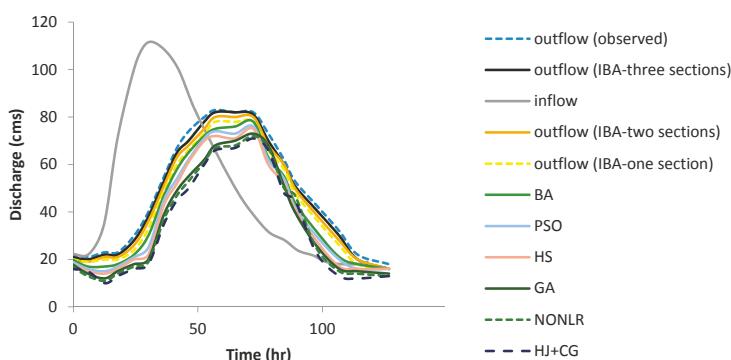
### 3.2. Multi-Interval Flood Routing (Wilson Flood)

It can be seen from Table 5 that SSD, SAD, and MARE for IBA, BA, GA, and PSO for three reaches were less than for two reaches, and their values for two reaches were less than for the single reach. The best results by IBA were obtained for three reach flood routing. For flood routing using single, two, and three reaches, all evolutionary algorithms predicted the peak discharge accurately, such that the difference with the observed value was 0. For example, SSD for IBA, for one reach was 4.123, whilst it was 3.988 for three reaches. SSD and SAD for IBA for one, two, and three reaches were 1 and 85%, compared to other algorithms, and thus, IBA better performed than other algorithms. The results based on Varex Q showed that the generated hydrograph based on IBA with consideration of more value for Varex Q had a better performance than the other algorithms, and it had a high match with the observed hydrograph. The investigation of Ep showed the value of the index had decreased from 50% to 97% based on IBA, compared to the other methods for two and three intervals. In addition, the value of MARE had decreased from 6% to 56% based on IBA, for the two and three reaches. Furthermore, the agreement index (d) showed a better performance for IBA, achieving a value closer to 1 compared to the other algorithms.

An increase in the number of reaches for evolutionary algorithms increased the accuracy of flood routing. Figure 3 shows the performance of IBA for flood routing with one, two, and three reaches, which is more consistent with observed discharges. Moreover, the computational time showed better performance for the IBA, compared to the other algorithms. For example, the computational time for the IBA based on one reach was 5 s, whilst it was 7, 8, and 9 s for the BA, PSO, and GA, respectively.

**Table 4.** Computed error indexes for Wilson flood.

Method	K	X	m	SSD	SAD	EP	ETP	MARE	VarexQ
SLSM	0.0010	0.2500	2.3470	143.600	46.40	0.0216	0	0.0561	98.33
HJ + CG	0.0069	0.2685	1.9291	49.640	25.20	0.0059	0	0.0301	99.59
HJ + DFP	0.0764	0.2677	1.8987	45.640	24.80	0.0035	0	0.0331	99.63
NONLR	0.0600	0.2700	2.3600	41.280	25.20	0.0083	1	0.0251	99.60
GA	0.1033	0.2873	1.8282	39.230	23.80	0.0082	0	0.0311	99.70
HS	0.0833	0.2873	1.8630	36.780	23.40	0.0107	0	0.0312	99.63
PSO	0.0755	0.2981	3.681	8.820	9.771	0.0005	0	0.0261	99.93
PS	0.4891	0.2714	1.8281	62.65	29.48	0.2901	0	0.0345	99.25
HMBO	0.6304	0.3399	1.8533	36.242	37.451	0.7001	0	0.0281	99.69
BA	0.0311	0.2934	0.8235	5.123	8.112	0.0004	0	0.0312	99.96
Present study IBA	0.0312	0.2997	1.8678	4.123	7.112	0.0002	0	0.0245	99.98



**Figure 3.** The simulated Hydrographs for Wilson flood using different methods.

**Table 5.** Flood routing for Wilson flood according to three different intervals.

Method	K	X	m	$\beta$	SSD	EP	MARE	VarexQ	Time (s)	d
IBA	0.0312	0.2997	1.8678	$\beta_1 = 0.0212$ $\beta_2 = 0.0114$	4.123	7.112	0.0002	0.0245	99.98	5
BA	0.0314	0.2996	1.8923	$\beta_1 = 0.0210$ $\beta_2 = 0.0114$	5.123	8.125	0.0004	0.0312	99.96	7
PSO	0.0755	0.2981	3.681	$\beta_1 = 0.0199$ $\beta_2 = 0.0065$	8.820	9.771	0.0005	0.0261	99.93	8
GA	0.1033	0.2873	1.8282	$\beta_1 = 0.0111$ $\beta_2 = 0.0033$	39.230	23.80	0.0082	0.0311	99.70	9
IBA	$K_1 = 0.0378$ $K_2 = 0.0345$	$X_1 = 0.2267$ $X_2 = 0.2245$	$m_1 = 1.9435$ $m_2 = 1.8912$	$\beta_1 = 0.0124$ $\beta_2 = 0.0114$	4.011	7.011	0.0002	0.0231	99.98	8
BA	$K_1 = 0.0368$ $K_2 = 0.0355$	$X_1 = 0.2167$ $X_1 = 0.2457$	$m_1 = 1.9735$ $m_2 = 1.8812$	$\beta_1 = 0.0021$ $\beta_2 = 0.054$	4.021	7.105	0.0003	0.0241	99.97	10
PSO	$K_1 = 0.0871$ $K_2 = 0.0881$	$X_1 = 0.2676$ $X_2 = 0.2512$	$m_1 = 1.2311$ $m_2 = 1.2212$	$\beta_1 = 0.0034$ $\beta_2 = 0.045$	8.123	9.123	0.0004	0.0251	99.93	12
GA	$K_1 = 0.0861$ $K_2 = 0.0882$	$X_1 = 0.2214$ $X_2 = 0.2312$	$m_1 = 1.1211$ $m_2 = 1.1112$	$\beta_1 = 0.0074$ $\beta_2 = 0.054$	38.11	22.121	0.0082	0.0281	99.70	14
IBA	$K_1 = 0.0871$ $K_2 = 0.0851$ $K_3 = 0.0812$	$X_1 = 0.2876$ $X_2 = 0.2745$ $X_3 = 0.2212$	$m_1 = 2.0121$ $m_2 = 2.111$ $m_3 = 2.123$	$\beta_1 = 0.0174$ $\beta_1 = 0.0162$ $\beta_1 = 0.0151$	3.988	6.989	0.0001	0.0221	99.98	15
BA	$K_1 = 0.0841$ $K_2 = 0.0852$ $K_3 = 0.0822$	$X_1 = 0.2976$ $X_2 = 0.2641$ $X_3 = 0.2314$	$m_1 = 2.1122$ $m_2 = 2.221$ $m_3 = 2.2231$	$\beta_1 = 0.0139$ $\beta_1 = 0.0111$ $\beta_1 = 0.0167$	4.001	6.999	0.0002	0.0231	99.97	16
PSO	$K_1 = 0.078$ $K_2 = 0.0812$ $K_3 = 0.0816$	$X_1 = 0.4567$ $X_2 = 0.4569$ $X_3 = 0.4745$	$m_1 = 5.112$ $m_2 = 5.114$ $m_3 = 5.116$	$\beta_1 = 0.0094$ $\beta_1 = 0.0082$ $\beta_1 = 0.0065$	7.126	8.989	0.0003	0.0241	99.94	19
GA	$K_1 = 0.0651$ $K_2 = 0.0612$ $K_3 = 0.0724$	$X_1 = 0.3212$ $X_2 = 0.3414$ $X_3 = 0.3515$	$m_1 = 6.123$ $m_2 = 6.178$ $m_3 = 6.115$	$\beta_1 = 0.0056$ $\beta_1 = 0.0065$ $\beta_1 = 0.0033$	37.123	21.123	0.0072	0.0271	99.72	22

### 3.3. Karahan Flood

Karahan et al. [22] routed a flood using various algorithms. This study considered the 1960 flood on the River Wye, UK. River Wye is 69.75 km from Erwood to Belmont with consideration of lateral inflow, which was ignored in the previous studies, as shown in Table 7. In this study, the proposed model was evaluated considering the flood routing based on lateral inflow, which occurred in this Karahan flood. The data was extracted from Karahan [22].

This flood was also used in the present study. The population used for IBA was 50, the maximum frequency was 5, the minimum frequency was zero, and the maximum sound loudness was 95%. In addition, the number of chromosomes for GA was 50, the probability of mutation was 0.6, and the recombination rate was 0.7. Furthermore, the number of particles used in the particle swarm algorithm was 50, the inertia coefficient was 0.7, and the personal and global learning coefficients were 2. Table 6 shows a comparison of algorithms used in flood routing in a single reach. The value of SSD for IBA was 17,120.21, which had reduced by 81%, 87%, 84%, and 10% compared with PSO, GA, HS, BA, respectively. In addition, SAD for IBA was 117, which had reduced by 83%, 88%, 94%, and 12% compared to PSO, GA, HS, and BA, respectively. The other error indices also showed a more favorable performance of IBA, compared to other algorithms. The predicted peak discharge difference with observed discharge was 0.002 cm, which was less than the other algorithms (Table 6). The time difference between predicted and observed discharge peak time for IBA was one hour, whilst this time for other algorithms was 6 hours, so IBA had a better performance. In addition, VarexQ for IBA had a larger value than other methods, which indicated a better performance. The difference of peak discharge based on IBA with observed peak was 69 cms (the closest value to the observed one), whilst it was 135 cms, 134 cms, and 70 cms for HS, PSO, and BA, respectively.

Table 7 compares IBA, BA, GA, and PSO in flood routing (Karahan flood) for single, two, and three reaches. SSD for IBA for three reaches was 16,098.21, which was 6 percent lower than for a single reach with IBA. Moreover, SAD for IBA for three reaches was 102, which had decreased by 12% compared to a single reach. Thus, IBA had an improved performance in routing with three reaches relative to single and two intervals. This was also true for the other algorithms, as shown in Table 5. Figure 4 also shows the superior performance of IBA based on three reaches. SSD using IBA for three reaches was reduced by 46%, 51%, and 5.3% compared to GA, PSO, and BA, respectively. Furthermore, SAD using IBA for three, two, and one reaches was reduced by 23–89%, compared to the other algorithms. As a result, IBA had a better performance than BA, GA, and PSO because the error indexes had the lowest value using the IBA, compared to the other algorithms. Examining the computation time showed that IBA achieved the optimal value of the objective function “under any number of multi-reach interval” faster than the other algorithms. The lowest value of the MARE index was the one associated with the IBA, on the other hand, this value decreased by 12% and 91% when using two and three reaches, respectively. Furthermore, the d index showed the IBA algorithm and simulated hydrographs using IBA had a better match with the observed hydrograph. Many studies did not consider investigate whether parameters generated by calibration could show different floods on the same river reach. In this article, this issue was considered. The optimal parameters based on IBA and three intervals for the flood event in December 1960, were used to obtain the output hydrograph for a flood on the same river on the another flood event in January 1969, based on the inflow hydrograph on January 1969, and it was considered as the first scenario. Then, the output hydrograph was extracted based on application of obtained parameters of Muskingum models for January 1969. It meant that the model was calibrated with the 1960 storm “first scenario”.

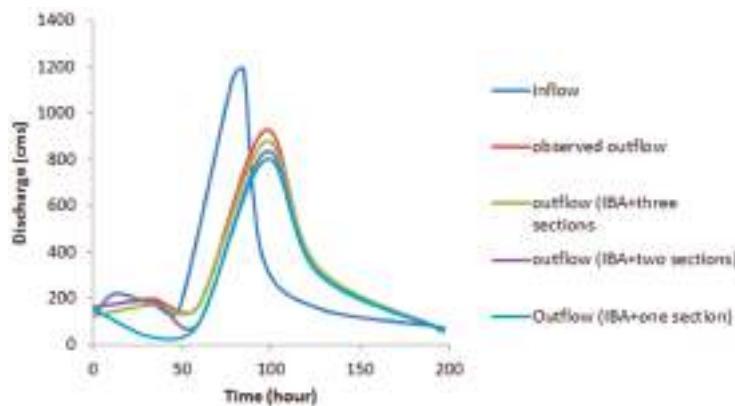


Figure 4. Simulated hydrograph for Karahan flood.

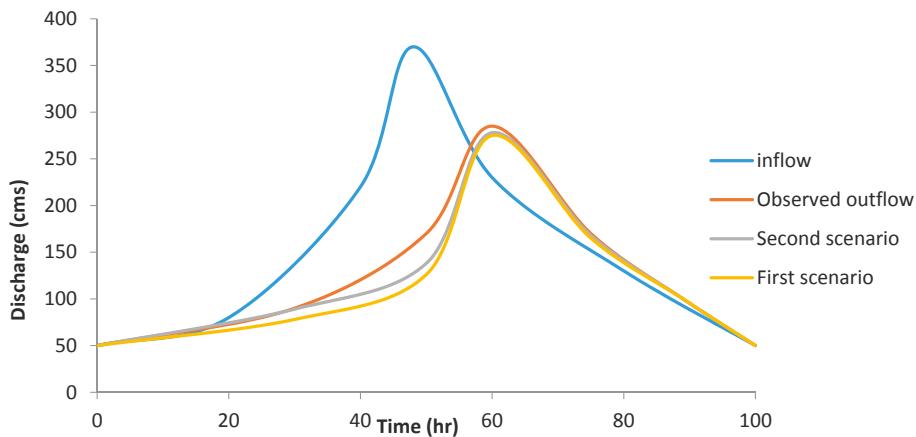
Table 6. Inflow and outflow for Karahan flood.

Time (h)	Inflow (cms)	Observed Outflow (cms)	HS [2,25]	GA	PSO	BA	Present Study IBA
0	154	102	154	132	102	102	102
6	150	140	154	152.21	154	137.89	137.24
12	219	169	152	153.44	152.1	165.78	166.12
18	182	190	181	178.11	179.4	185.43	186.11
24	182	209	191	190.45	190.9	209.01	207.12
30	192	218	185	185.1	185.4	212.32	214.33
36	165	210	187	188.21	186.9	204.45	205.24
42	150	194	179	179.45	180.20	191.32	192.12
48	128	172	162	163.11	164.10	10.45	171.25
54	168	149	141	142.11	143.70	141.44	141.38
60	260	136	154	151.12	152.8	132.22	133.56
66	471	228	198	197.11	196.3	221.14	222.21
72	717	303	264	265.21	267.3	299.12	301.12
78	1092	366	344	349.10	351.4	387.12	385.21
84	1145	456	416	423.11	431.8	451.22	453.12
90	600	615	599	600.12	617.4	610.34	611.21
96	365	830	871	872.32	881.5	826.34	827.12
102	277	969	834	835.11	836.6	899.34	900.12
108	277	665	689	690.11	696.2	667.24	665.21
114	187	519	535	534.11	549.2	522.34	520.21
120	161	444	397	400.1	416.8	455.67	453.11
126	143	321	283	287.10	305.10	314.32	316.11
132	126	208	202	203.11	221.4	212.22	210.25
138	115	176	152	155.21	164.9	177.54	170.10
144	102	148	124	131.10	131.20	151.23	145.11
150	93	125	106	108.12	110.0	127.34	119.14
156	88	114	94	106.21	96.04	116.34	112.10
162	82	106	88	88.23	89.20	107.21	105.10
168	76	97	82	81.21	82.70	92.12	93.43
174	73	89	75	76.11	76.30	91.23	88.11
180	70	81	73	73.10	73.10	82.34	80.21
186	67	76	69	69	69.80	78.12	75.10
192	63	71	66	66	66.7	72.34	69.21
198	59	66	62	62	62.40	65.21	64
SSD	-	-	37,944.14	32,944.14	31,099.52	19,122.23	17,120.21
SAD			2162	1012	695	134	117
EP			0.278	0.078	0.090	0.068	0.002
ETP			6	6	6	1	1
MARE			0.33	0.10	0.09	0.02	0.01
VarexQ			83.29	84.78	98.05	98.12	99.15

Table 7. Results of four models (i.e., IBA, BA, PSO and GA) for Karahan flood according to three different intervals.

Method	K	X	m	$\beta$	SSD	SAD	EP	ETP	MARE	VareQ	Time (s)	d
<b>One section</b>												
<b>Two sections</b>												
<b>Three sections</b>												
IBA	0.612	0.401	1.633	0.0142	17,120.21	117	0.002	1	0.01	99.15	6	0.94
BA	0.616	0.422	1.654	0.0136	19,122.23	134	0.068	1	0.02	98.12	9	0.93
PSO	0.586	0.365	1.545	0.0138	31,099.51	695	0.090	6	0.09	98.05	10	0.90
GA	0.456	0.322	1.824	0.0137	32,944.14	1012	0.078	6	0.10	94.078	12	0.89
IBA	$K_1 = 0.672$ $K_2 = 0.524$	$X_1 = 0.382$ $X_2 = 0.375$	$m_1 = 1.723$ $m_2 = 1.645$	$\beta_1 = 0.0174$ $\beta_2 = 0.0154$	17,091.20	115	0.002	1	0.01	99.25	8	0.93
BA	$K_1 = 0.652$ $K_2 = 0.521$	$X_1 = 0.352$ $X_2 = 0.355$	$m_1 = 1.623$ $m_2 = 1.642$	$\beta_1 = 0.0214$ $\beta_2 = 0.0124$	17,114.25	122	0.058	1	0.01	99.15	11	0.90
PSO	$K_1 = 0.112$ $K_2 = 0.110$	$X_1 = 0.289$ $X_2 = 0.284$	$m_1 = 1.623$ $m_2 = 1.545$	$\beta_1 = 0.0064$ $\beta_2 = 0.0039$	30,235.45	687	0.088	6	0.08	98.12	14	0.89
GA	$K_1 = 0.78$ $K_2 = 0.689$	$X_1 = 0.244$ $X_2 = 0.232$	$m_1 = 1.611$ $m_2 = 1.811$	$\beta_1 = 0.0027$ $\beta_2 = 0.0029$	31,231.23	1009	0.068	6	0.10	94.79	16	0.88
IBA	$K_1 = 0.692$ $K_2 = 0.690$ $K_3 = 0.612$	$X_1 = 0.392$ $X_2 = 0.391$ $X_3 = 0.394$	$m_1 = 1.112$ $m_2 = 1.114$ $m_3 = 1.116$	$\beta_1 = 0.0178$ $\beta_2 = 0.0162$ $\beta_3 = 0.0144$	16,098.21	102	0.002	1	0.008	99.56	10	0.91
BA	$K_1 = 0.694$ $K_2 = 0.696$ $K_3 = 0.622$	$X_1 = 0.394$ $X_2 = 0.399$ $X_3 = 0.394$	$m_1 = 1.115$ $m_2 = 1.117$ $m_3 = 1.116$	$\beta_1 = 0.0074$ $\beta_2 = 0.0124$ $\beta_3 = 0.0134$	16,999.21	108	0.0038	1	0.009	99.54	15	0.86
PSO	$K_1 = 0.237$ $K_2 = 0.312$ $K_3 = 0.321$	$X_1 = 0.298$ $X_2 = 0.321$ $X_3 = 0.312$	$m_1 = 1.311$ $m_2 = 1.411$ $m_3 = 1.512$	$\beta_1 = 0.0044$ $\beta_1 = 0.0029$ $\beta_3 = 0.0020$	30,230.21	667	0.081	6	0.06	99.11	17	0.84
GA	$K_1 = 0.900$ $K_2 = 0.878$ $K_3 = 0.815$	$X_1 = 0.296$ $X_2 = 0.294$ $X_3 = 0.224$	$m_1 = 1.655$ $m_2 = 1.652$ $m_3 = 1.651$	$\beta_1 = 0.0014$ $\beta_2 = 0.0012$ $\beta_3 = 0.0010$	30,298.11	987	0.057	6	0.09	96.12	19	0.82

The value of SSD for the first scenario was 878 and the peak value was 280 cms, whilst the observed peak value was 285 and there was a small difference between the observed and the simulated peak value (Figure 5). The SSD for the second scenario was 869 and the peak value was 282 cms. Although, the second scenario acted better than the first scenario, the results for the first scenario were acceptable. The results were so similar because of the accurate sensitivity analysis considered for the Muskingum model for the objective function and different parameters, such as in previous sections. Although the proposed model structure successfully predicted the flood routing, further research could consider different Muskingum models based on more parameters to investigate the performance of IBA. In addition, application of these models helps the hydraulic designers to have the accurate value for the peak discharge and the flood characteristics, to be able to optimally design the target structure.



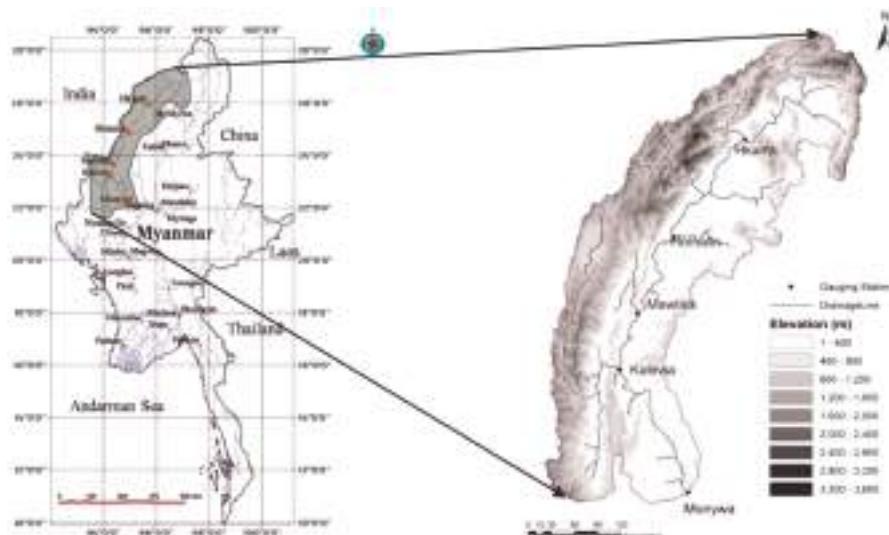
**Figure 5.** The first and second scenario for calibration of model.

### 3.4. Chindwin River

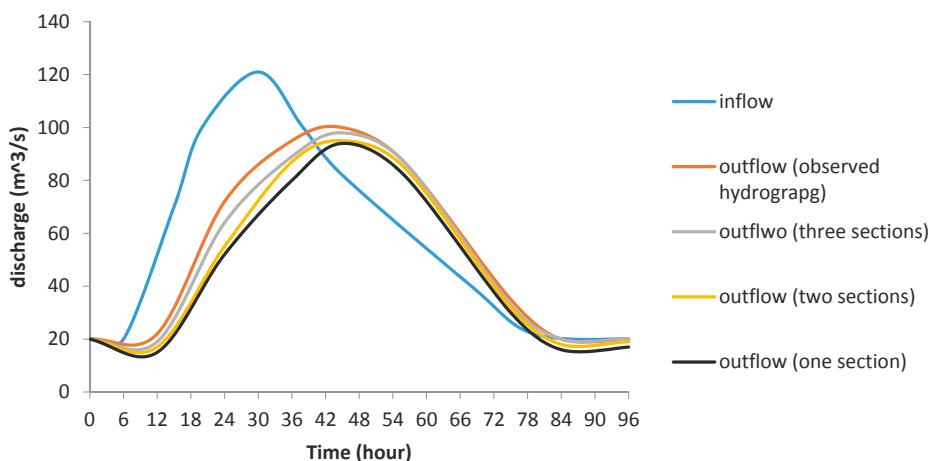
One of the most important Myanmar Rivers is the Chindwin River, shown in Figure 6. This river, over the past twenty years, has experienced many floods. Due to severe flood conditions, the construction of hydraulic structures and living conditions in the downstream areas are difficult and there is high lateral inflow for this basin. Therefore, it is important to predict river flood conditions. The length of the river is 114 km and the surface of the basin is  $7485 \text{ km}^2$ . A historical flood considered in this study occurred in July 2004, and the Mawliak Station recording the flood is shown in Figure 5. Table 8 shows a comparison of different methods for the first flood with flood routing using a single reach. Comparing IBA, BA, GA, and PSO based on routing with a single reach, SSD for IBA relative to GA, BA, and PSO was reduced by 32%, 11%, and 42%, respectively. For routing with two reaches, SSD for IBA was reduced by 33%, 34%, and 12% compared to PSO, GA, and BA, respectively. For three reaches, SSD for IBA was reduced by 44%, 50%, and 37% compared to PSO, GA and BA, respectively. The SAD index for the three routings indicated the superiority of IBA. Comparison of routing with three, two, and one reaches indicated that three reaches improved routing. SSD in relation to routing with two and one reaches had reduced by 39% and 20%, respectively. ETp showed that there was no time lag in forecasting of peak flow by different algorithms. VarexQ showed that three-reach routing was better than two and one reach routings. Additionally, MARE for IBA-based routing had reduced by 64% and 62%, relative to one and two reach routing, respectively. Figure 7 shows that the three reach flood routing was superior to that based on two and single reaches. The results indicated that the IBA has the less computational time compared to the other algorithms, as shown in Table 6. In addition, considering the attained value of the d index, it was obvious that IBA had the best performance compared to the other algorithms.

Table 8. The computed error indexes for Myanmar River.

Method	K	X	m	$\beta$	SSE	SAD	EP	ETP	MARE	VareQ	Time (s)	d
<b>One section</b>												
<b>Two sections</b>												
<b>Three sections</b>												
IBA	0.411	0.301	1.611	0.0344	8.24	2.12	0.002	0	0.015	99.22	7	0.93
BA	0.410	0.304	1.612	0.321	9.22	3.14	0.004	0	0.017	99.16	9	0.92
PSO	0.386	0.255	1.545	0.0265	12.22	3.24	0.078	0	0.095	98.15	10	0.89
GA	0.256	0.312	1.524	0.0222	14.25	4.25	0.089	0	0.102	94.78	11	0.87
IBA	$K_1 = 0.472$ $K_2 = 0.424$	$X_1 = 0.392$ $X_2 = 0.365$	$m_1 = 1.721$ $m_2 = 1.635$	$\beta_1 = 0.0212$ $\beta_2 = 0.0139$	7.99	2.10	0.002	0	0.014	99.25	10	0.91
BA	$K_1 = 0.479$ $K_2 = 0.428$	$X_1 = 0.394$ $X_2 = 0.368$	$m_1 = 1.821$ $m_2 = 1.433$	$\beta_1 = 0.0210$ $\beta_2 = 0.0131$	9.11	2.89	0.003	0	0.016	99.18	12	0.90
PSO	$K_1 = 0.102$ $K_2 = 0.110$	$X_1 = 0.279$ $X_2 = 0.224$	$m_1 = 1.622$ $m_2 = 1.542$	$\beta_1 = 0.0197$ $\beta_1 = 0.0190$	11.95	3.09	0.088	0	0.088	98.22	14	0.89
GA	$K_1 = 0.78$ $K_2 = 0.789$	$X_1 = 0.244$ $X_2 = 0.212$	$m_1 = 1.610$ $m_2 = 1.611$	$\beta_1 = 0.0196$ $\beta_2 = 0.0188$	12.24	4.55	0.068	0	0.100	94.89	16	0.87
IBA	$K_1 = 0.492$ $K_2 = 0.491$ $K_3 = 0.512$	$X_1 = 0.292$ $X_2 = 0.261$ $X_3 = 0.294$	$m_1 = 1.110$ $m_2 = 1.112$ $m_3 = 1.114$	$\beta_1 = 0.0342$ $\beta_2 = 0.0288$ $\beta_3 = 0.0488$	5.12	1.98	0.002	0	0.005	99.56	18	0.91
BA	$K_1 = 0.412$ $K_2 = 0.471$ $K_3 = 0.514$	$X_1 = 0.291$ $X_2 = 0.254$ $X_3 = 0.292$	$m_1 = 1.112$ $m_2 = 1.114$ $m_3 = 1.116$	$\beta_1 = 0.0342$ $\beta_2 = 0.0148$ $\beta_3 = 0.0138$	8.11	2.10	0.005	0	0.003	99.41	20	0.90
PSO	$K_1 = 0.236$ $K_2 = 0.311$ $K_3 = 0.319$	$X_1 = 0.295$ $X_2 = 0.320$ $X_3 = 0.310$	$m_1 = 0.211$ $m_2 = 0.221$ $m_3 = 0.212$	$\beta_1 = 0.0221$ $\beta_2 = 0.0212$ $\beta_2 = 0.0217$	9.27	2.12	0.081	0	0.0612	99.31	22	0.87
GA	$K_1 = 0.910$ $K_2 = 0.876$ $K_3 = 0.815$	$X_1 = 0.396$ $X_2 = 0.396$ $X_3 = 0.324$	$m_1 = 1.655$ $m_2 = 1.652$ $m_3 = 1.651$	$\beta_1 = 0.0251$ $\beta_2 = 0.0218$ $\beta_2 = 0.0217$	10.12	3.25	0.057	0	0.090	96.18	24	0.85

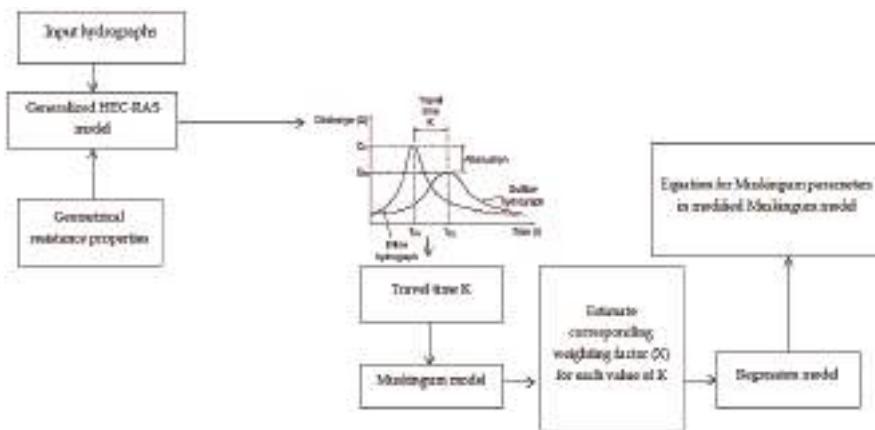


**Figure 6.** The location of the River in Myanmar.



**Figure 7.** Simulated hydrograph for Myanmar River using IBA.

To enhance further, the used structure of the proposed Muskingum model coupled with IBA showed the highest abilities, for studying the flood plain with back water. Furthermore, such enhancement for the Muskingum model could be successfully applied as an easy and inexpensive method for computing the time and shape, for an overbank flood, when there is back water and inertia influences along a river channel. In fact, with the new nature-inspired optimization algorithms, the traditional Muskingum method could be integrated with the hydrodynamic software packages, such as HEC-RAS model. The development procedure could be carried out by utilizing HEC-RAS software as the model input for hydrographs and geometric characteristic, to estimate the travel times and attenuations peak, whilst the weighted coefficient (X) value could be achieved based on the optimization model and Muskingum model, as shown in Figure 8. As a result, the proposed model in this research showed the potential to be suitable and appropriate for studying and analyzing flood propagation and flood mapping.



**Figure 8.** Advanced Muskingum model for flood plain with complex boundary condition.

#### 4. Conclusions

The present study investigated the potential of utilizing a three-parameter Muskingum model coupled with Improved Bat Algorithm (IBA) to accurately predict flood routing. Three different case studies have been used in this study, to evaluate the performance of the proposed model. These three case studies were the Wilson flood, Karahan flood, and Myanmar River. Seven different performance indices were used to examine and compare the performance of the proposed model over other algorithms. In addition, discretization of the river stream was considered to improve model accuracy. The results showed that IBA outperformed all other algorithms and was able to reduce the SSD by percentages ranging between 20% and 84%, compared with the other algorithms. In addition, the achieved results using the IBA could predict the peak discharge accurately, with a value very close to the observed one. Under the Karahan flood, IBA considerably achieved the minimum level of error indices for a single reach, compared to other algorithms. Finally, IBA in flood routing with three intervals had a better performance than with single and two reaches. The division of the river into different reaches increased the accuracy of flood routing. The performance of IBA for the river in Myanmar, also showed that the simulated hydrograph with three reaches was more accurate. For example, the computational time for the IBA based on three intervals was 2, 4, and 6 s less than, BA, PSO, and GA, respectively. Furthermore, the EP for the IBA was 33%, 96%, and 97% less than BA, PSO, and GA, respectively. As a result, the proposed Muskingum model coupled with the IBA, could be considered as a strong alternative method for predicting flood routing characteristics.

**Author Contributions:** S.F., H.K., A.E.S., O.K. and V.P.S. initiated the research point of the hydrological problem and supervised the study. N.F. collected the data and performed the modeling. M.E. and M.F.A. handled the writing up of the introduction and methodology. The analysis of the modeling outcomes have been handled by N.S.M., N.F. organized the whole manuscript and managed the paper submission and revision.

**Funding:** This research was funded by the University of Malaya Research Grant (UMRG) coded RP025A-18SUS University of Malaya, Malaysia.

**Acknowledgments:** The authors would like to thank so much the data supplier. We also thank all reviewers and the Editor-In-Chief for their insightful comments that have improved the quality of the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

IBA	Improved Bat Algorithm
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
PS	Pattern Search
HS	Harmony Search
HBMO	Honey Bee Mating Optimization
NMSA	Nelder-Mead Simplex Algorithm
GPA	Genetic Programming Algorithm
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
HA	Harmony Algorithm
FLA	Frog Leaping Algorithm
(IHBA)	Improved Honey Bee Algorithm
(IWOA)	Invasive Weed Optimization Algorithm
(BA)	Bat Algorithm
BI	Bat Intelligence
TPMM	Three-Parameter Muskingum method
SSD	Sum of Squares
SAD	Sum of Absolute Deviations
EP	Error of Peak
ETP	Error of Time to Peak

## References

- Barati, R.; Badfar, M.; Azizyan, G.; Akbari, G.H. Discussion of “Parameter Estimation of Extended Nonlinear Muskingum Models with the Weed Optimization Algorithm” by Farzan Hamed, Omid Bozorg-Haddad, Maryam Pazoki, Hamid-Reza Asgari, Mehran Parsa, and hugo a. Loáiciga. *J. Irrig. Drain. Eng.* **2018**, *144*, 7017021. [[CrossRef](#)]
- Hamed, F.; Bozorg-Haddad, O.; Pazoki, M.; Asgari, H.R.; Parsa, M.; Loáiciga, H.A. Parameter estimation of extended nonlinear muskingum models with the weed optimization algorithm. *J. Irrig. Drain. Eng.* **2016**, *142*, 4016059. [[CrossRef](#)]
- Zhang, S.; Kang, L.; Conservation, B.Z. Parameter estimation of nonlinear muskingum model with variable exponent using adaptive genetic algorithm. In *Environmental Conservation, Clean Water, Air & Soil (CleanWAS)*; IWA Publishing: London, UK, 2017; pp. 231–234.
- Bradbrook, K.F.; Lane, S.N.; Waller, S.G.; Bates, P.D. Two dimensional diffusion wave modelling of flood inundation using a simplified channel representation. *Int. J. River Basin Manag.* **2004**, *2*, 211–223. [[CrossRef](#)]
- Neal, J.; Villanueva, I.; Wright, N.; Willis, T.; Fewtrell, T.; Bates, P. How much physical complexity is needed to model flood inundation? *Hydrol. Process.* **2012**, *26*, 2264–2282. [[CrossRef](#)]
- Hunter, N.M.; Bates, P.D.; Neelz, S.; Pender, G.; Villanueva, I.; Wright, N.G.; Liang, D.; Falconer, R.A.; Lin, B.; Waller, S.; et al. Benchmarking 2D hydraulic models for urban flooding simulations. *Proc. Inst. Civ. Eng. Water Manag.* **2008**, *161*, 13–30. [[CrossRef](#)]
- Dottori, F.; Todini, E. Testing a simple 2D hydraulic model in an urban flood experiment. *Hydrol. Process.* **2013**, *27*, 1301–1320. [[CrossRef](#)]
- Kim, B.; Sanders, B.F.; Schubert, J.E.; Famiglietti, J.S. Mesh type tradeoffs in 2D hydrodynamic modeling of flooding with a godunov-based flow solver. *Adv. Water Resour.* **2014**, *68*, 42–61. [[CrossRef](#)]
- Prestininzi, P. Suitability of the diffusive model for dam break simulation: Application to a cadam experiment. *J. Hydrol.* **2008**, *361*, 172–185. [[CrossRef](#)]
- Aricò, C.; Sinagra, M.; Begnudelli, L.; Tucciarelli, T. MAST-2D diffusive model for flood prediction on domains with triangular Delaunay unstructured meshes. *Adv. Water Resour.* **2011**, *34*, 1427–1449. [[CrossRef](#)]
- Schubert, J.E.; Sanders, B.F.; Smith, M.J.; Wright, N.G. Unstructured mesh generation and landcover-based resistance for hydrodynamic modeling of urban flooding. *Adv. Water Resour.* **2008**, *31*, 1603–1621. [[CrossRef](#)]

12. Bates, P.D.; Horritt, M.S.; Fewtrell, T.J. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *J. Hydrol.* **2010**, *387*, 33–45. [[CrossRef](#)]
13. Costabile, P.; Costanzo, C.; Macchione, F. Performances and limitations of the diffusive approximation of the 2-D shallow water equations for flood simulation in urban and rural areas. *Appl. Numer. Math.* **2017**, *116*, 141–156. [[CrossRef](#)]
14. Fassoni-Andrade, A.C.; Fan, F.M.; Collischonn, W.; Fassoni, A.C.; Paiva, R.C. Comparison of numerical schemes of river flood routing with an inertial approximation of the Saint Venant equations. *RBRH* **2018**, *23*, e10. [[CrossRef](#)]
15. Singh, V.P.; Aravamuthan, V. Errors of kinematic-wave and diffusion-wave approximations for steady-state overland flows. *Catena* **1996**, *27*, 209–227. [[CrossRef](#)]
16. Costabile, P.; Macchione, F.; Natale, L.; Petaccia, G. Flood mapping using LIDAR DEM. Limitations of the 1-D modeling highlighted by the 2-D approach. *Nat. Hazards* **2015**, *77*, 181–204. [[CrossRef](#)]
17. Chatila, J.G. Muskingum Method, EXTRAN and ONE-D for routing unsteady flows in open channels. *Can. Water Resour. J.* **2003**, *28*, 481–498. [[CrossRef](#)]
18. O’Sullivan, J.J.; Ahilan, S.; Bruen, M. A modified Muskingum routing approach for floodplain flows: Theory and practice. *J. Hydrol.* **2012**, *470*, 239–254.
19. Yoo, C.; Lee, J.; Lee, M. Parameter Estimation of the Muskingum Channel Flood-Routing Model in Ungauged Channel Reaches. *J. Hydrol. Eng.* **2017**, *22*, 5017005. [[CrossRef](#)]
20. Geem, Z.W. Parameter Estimation of the Nonlinear Muskingum Model Using Parameter-Setting-Free Harmony Search. *J. Hydrol. Eng.* **2011**, *16*, 684–688. [[CrossRef](#)]
21. Barati, R. Parameter estimation of nonlinear Muskingum models using Nelder-Mead simplex algorithm. *J. Hydrol. Eng.* **2011**, *16*, 946–954. [[CrossRef](#)]
22. Karahan, H.; Gurarslan, G.; Geem, Z.W. Parameter estimation of the nonlinear Muskingum flood-routing model using a hybrid harmony search algorithm. *J. Hydrol. Eng.* **2013**, *18*, 352–360. [[CrossRef](#)]
23. Orouji, H.; Bozorg Haddad, O.; Fallah-Mehdipour, E.; Mariño, M.A. Estimation of Muskingum parameter by meta-heuristic algorithms. *Proc. Inst. Civ. Eng. Water Manag.* **2013**, *166*, 315–324. [[CrossRef](#)]
24. Easa, S.M. New and improved four-parameter non-linear Muskingum model. *Proc. Inst. Civ. Eng. Water Manag.* **2014**, *167*, 288–298. [[CrossRef](#)]
25. Talatahari, S.; Sheikholeslami, R.; Farahmand Azar, B.; Daneshpajouh, H. Optimal Parameter Estimation for Muskingum Model Using a CSS-PSO Method. *Adv. Mech. Eng.* **2013**, *5*, 480954. [[CrossRef](#)]
26. Ouyang, A.; Li, K.; Truong, T.K.; Sallam, A.; Sha, E.H.-M. Hybrid particle swarm optimization for parameter estimation of Muskingum model. *Neural Comput. Appl.* **2014**, *25*, 1785–1799. [[CrossRef](#)]
27. Geem, Z.W. Issues in optimal parameter estimation for the nonlinear Muskingum flood routing model. *Eng. Optim.* **2014**, *46*, 328–339. [[CrossRef](#)]
28. Haddad, O.B.; Hamed, F.; Fallah-Mehdipour, E.; Orouji, H.; Mariño, M.A. Application of a hybrid optimization method in Muskingum parameter estimation. *J. Irrig. Drain. Eng.* **2015**, *141*, 4015026. [[CrossRef](#)]
29. Niazkar, M.; Afzali, S.H. Assessment of modified honey bee mating optimization for parameter estimation of nonlinear Muskingum models. *J. Hydrol. Eng.* **2015**, *20*, 4014055. [[CrossRef](#)]
30. Moghaddam, A.; Behmanesh, J.; Farsijani, A. Parameters estimation for the new four-parameter nonlinear muskingum model using the particle swarm optimization. *Water Resour. Manag.* **2016**, *30*, 2143–2160. [[CrossRef](#)]
31. Kang, L.; Zhang, S. Application of the elitist-mutated PSO and an improved CSA to estimate parameters of linear and nonlinear Muskingum flood routing models. *PLoS ONE* **2016**, *11*, e0147338. [[CrossRef](#)] [[PubMed](#)]
32. Lee, E.; Lee, H.; Kim, J. Development and Application of Advanced Muskingum Flood Routing Model Considering Continuous Flow. *Water* **2018**, *10*, 760. [[CrossRef](#)]
33. Wang, L.; Lapin, S.; Wu, J.Q.; Elliot, W.J.; Fiedler, F.R. Accuracy of the Muskingum-Cunge method for constant-parameter diffusion-wave channel routing with lateral inflow. *arXiv* **2018**.
34. Jamil, M.; Zepernick, H.J.; Yang, X.S. Multimodal Function Optimization Using an Improved Bat Algorithm in Noise-Free and Noisy Environments. In *Nature-Inspired Computing and Optimization*; Patnaik, S., Yang, X.-S., Nakamatsu, K., Eds.; Modeling and Optimization in Science and Technologies; Springer International Publishing: Basel, Switzerland, 2017; Volume 10, pp. 29–49.
35. Bozorg-Haddad, O.; Karimirad, I.; Seifollahi-Aghmuni, S.; Loáiciga, H.A. Development and application of the bat algorithm for optimizing the operation of reservoir systems. *J. Water Resour. Plan. Manag.* **2015**, *141*, 4014097. [[CrossRef](#)]

36. Ahmadianfar, I.; Adib, A.; Salarijazi, M. Optimizing multi-reservoir operation: hybrid of bat algorithm and differential evolution. *J. Water Resour. Plan. Manag.* **2016**, *142*, 5015010. [[CrossRef](#)]
37. Cai, X.; Wang, H.; Cui, Z.; Cai, J.; Xue, Y.; Wang, L. Bat algorithm with triangle-flipping strategy for numerical optimization. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 199–215. [[CrossRef](#)]
38. Chakri, A.; Khelif, R.; Benouaret, M.; Yang, X.S. New directional bat algorithm for continuous optimization problems. *Expert Syst. Appl.* **2017**, *69*, 159–175. [[CrossRef](#)]
39. Ehteram, M.; Allawi, M.; Karami, H.; Mousavi, S. Optimization of Chain-Reservoirs' Operation with a New Approach in Artificial Intelligence. *Water Resour.* **2017**, *31*, 2085–2104. [[CrossRef](#)]
40. Allawi, M.F.; Jaafar, O.; Ehteram, M.; Mohamad Hamzah, F.; El-Shafie, A. Synchronizing Artificial Intelligence Models for Operating the Dam and Reservoir System. *Water Resour. Manag.* **2018**, *32*, 3373–3389. [[CrossRef](#)]
41. Allawi, M.F.; Jaafar, O.; Mohamad Hamzah, F.; Abdullah, S.M.S.; El-shafie, A. Review on applications of artificial intelligence methods for dam and reservoir-hydro-environment models. *Environ. Sci. Pollut. Res.* **2018**, *25*, 13446–13469. [[CrossRef](#)] [[PubMed](#)]
42. Allawi, M.F.; Jaafar, O.; Mohamad Hamzah, F.; Ehteram, M.; Hossain, M.S.; El-Shafie, A. Operating a reservoir system based on the shark machine learning algorithm. *Environ. Earth Sci.* **2018**, *77*, 366. [[CrossRef](#)]
43. El-shafie, A.; Mukhlisin, M.; Najah, A.A.; Taha, M.R. Performance of artificial neural network and regression techniques for rainfall-runoff prediction. *Int. J. Phys. Sci.* **2011**, *6*, 1997–2003.
44. Wilson, E.M. *Engineering Hydrology*; Palgrave: London, UK, 1990; pp. 1–49.
45. Legates, D.R.; McCabe, G.J. Evaluating the use of “goodness-of-fit” Measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* **1999**, *35*, 233–241. [[CrossRef](#)]
46. Ehteram, M.; Binti Othman, F.; Mundher Yaseen, Z.; Abdulmohsin Afan, H.; Falah Allawi, M.; Najah Ahmed, A.; Shahid, S.; Singh, V.P.; El-Shafie, A. Improving the Muskingum flood routing method using a hybrid of particle swarm optimization and bat algorithm. *Water* **2018**, *10*, 807. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# New Hybrids of ANFIS with Several Optimization Algorithms for Flood Susceptibility Modeling

Dieu Tien Bui <sup>1,2</sup>, Khabat Khosravi <sup>3</sup>, Shaojun Li <sup>4,\*</sup>, Himan Shahabi <sup>5,\*</sup>, Mahdi Panahi <sup>6</sup>, Vijay P. Singh <sup>7</sup>, Kamran Chapi <sup>8</sup>, Ataollah Shirzadi <sup>8</sup>, Somayeh Panahi <sup>6</sup>, Wei Chen <sup>9</sup> and Baharin Bin Ahmad <sup>10</sup>

- <sup>1</sup> Geographic Information Science Research Group, Ton Duc Thang University, Ho Chi Minh City, Vietnam; buitiendieu@tdt.edu.vn
- <sup>2</sup> Faculty of Environment and Labour Safety, Ton Duc Thang University, Ho Chi Minh City, Vietnam
- <sup>3</sup> Department of Watershed Management Engineering, Faculty of Natural Resources, Sari Agricultural Science and Natural Resources University, Sari 48181-68984, Iran; khabat.khosravi@gmail.com
- <sup>4</sup> State Key Laboratory of Geomechanics and Geotechnical Engineering, Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan 430071, China
- <sup>5</sup> Department of Geomorphology, Faculty of Natural Resources, University of Kurdistan, Sanandaj 66177-15175, Iran
- <sup>6</sup> Young Researchers and Elites Club, North Tehran Branch, Islamic Azad University, Tehran 19585-466, Iran; panahi2012@yahoo.com (M.P.); somipan@yahoo.com (S.P.)
- <sup>7</sup> Department of Biological and Agricultural Engineering & Zachry Department of Civil Engineering, Texas A & M University, College Station, TX 77843-2117, USA; vsingh@tamu.edu
- <sup>8</sup> Department of Rangeland and Watershed Management, Faculty of Natural Resources, University of Kurdistan, Sanandaj 66177-15175, Iran; k.chapi@uok.ac.ir (K.C.); a.shirzadi@uok.ac.ir (A.S.)
- <sup>9</sup> College of Geology & Environment, Xi'an University of Science and Technology, Xi'an 710054, China; chenwei.0930@163.com
- <sup>10</sup> Department of Geoinformation, Faculty of Geoinformation and Real Estate, Universiti Teknologi Malaysia, Skudai 81310, Malaysia; baharinahmad@utm.my
- \* Correspondence: sjli@whrsm.ac.cn (S.L.); h.shahabi@uok.ac.ir (H.S.); Tel.: +98-87-33-664-600 (H.S.)

Received: 6 August 2018; Accepted: 27 August 2018; Published: 7 September 2018

**Abstract:** This study presents three new hybrid artificial intelligence optimization models—namely, adaptive neuro-fuzzy inference system (ANFIS) with cultural (ANFIS-CA), bees (ANFIS-BA), and invasive weed optimization (ANFIS-IWO) algorithms—for flood susceptibility mapping (FSM) in the Haraz watershed, Iran. Ten continuous and categorical flood conditioning factors were chosen based on the 201 flood locations, including topographic wetness index (TWI), river density, stream power index (SPI), curvature, distance from river, lithology, elevation, ground slope, land use, and rainfall. The step-wise weight assessment ratio analysis (SWARA) model was adopted for the assessment of relationship between flood locations and conditioning factors. The ANFIS model, based on SWARA weights, was employed for providing FSMs with three optimization models to enhance the accuracy of prediction. To evaluate the model performance and prediction capability, root-mean-square error (RMSE) and receiver operating characteristic (ROC) curve (area under the ROC (AUROC)) were used. Results showed that ANFIS-IWO with lower RMSE (0.359) had a better performance, while ANFIS-BA with higher AUROC (94.4%) showed a better prediction capability, followed by ANFIS-IWO (0.939) and ANFIS-CA (0.921). These models can be suggested for FSM in similar climatic and physiographic areas for developing measures to mitigate flood damages and to sustainably manage floodplains.

**Keywords:** flood susceptibility modeling; ANFIS; cultural algorithm; bees algorithm; invasive weed optimization; Haraz watershed

## 1. Introduction

Floods that occur in a short duration with high peak discharge [1] are considered as the worst weather-related natural hazard worldwide causing huge losses of life and property as well as deep social impacts [2–4]. Floods are more hazardous than other natural catastrophic hazards, because they affect more than 20000 lives per year and adversely impact nearly 75 million people worldwide, especially through social impacts such as homelessness, major changes in human lives, and the environment [5,6]. In Asia, more than 50% of economic damages as well as over 90% of all human deaths are caused due to floods [7,8].

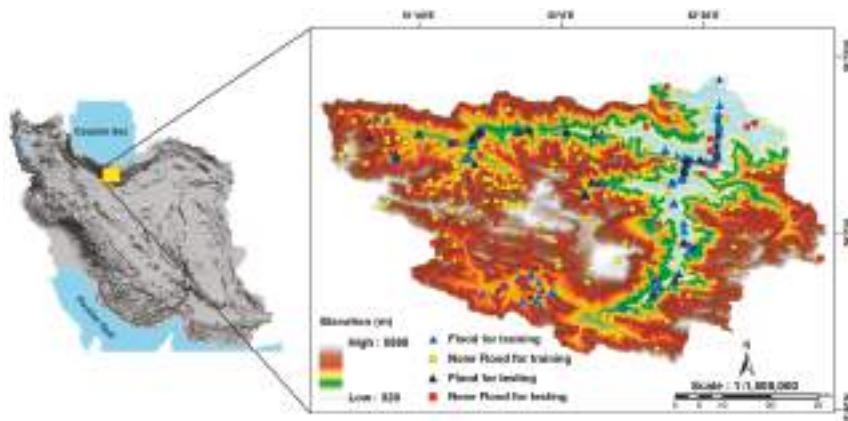
Many floods have recently occurred in Iran, especially in the northern parts, such as floods in 2012 at Noshahr, in 2013 at Neka and Behshahr, in 2013 and 2015 at Sari, and 2016 at Noshahr, which caused huge financial and human losses [9,10]. The frequencies and damages of these floods may increase in future due to severe climate change and extensive land-use changes in the country [11]. It is noted that a complete flood prevention is not possible; however, its spatial prediction can help mitigate its human and socio-economic losses [12]. Thus, one of the key points in flood management plans is the identification of flood prone areas so that damages can more likely be reduced by avoiding more construction and physical development in these regions. Therefore, it seems logical to seek approaches that can more and more accurately detect flood prone areas within watersheds. Flood measurement and modeling have been always considered two options for such an identification [13]. Since measurement of flood characteristics specifically during the event is hard, costly, and time consuming [14], modeling has been extensively used by scientists especially from the age of digital hydrology. A vast array modeling approaches, ranging from simple linear and empirical models to sophisticated non-linear physically-based models, have been used for flood simulations since then; however, a comprehensive and integrated flood modeling approach has not been achieved yet, due to the complexity, non-linearity, and dynamic structure of floods and their watersheds. Thus, the issue of flood occurrence forecasting and its mapping using physically-based rainfall-runoff models has remained as a challenge [15,16]. The weakness of one-dimensional, linear, and empirical hydrological methods is that watershed river morphology is not stable, and has dynamic characteristics, and they also take a black box view to the watershed such that they cannot provide any insight into the flooding process [17]. These models are also unable to depict rapid watershed responses [18–20]. Even though physically-based models can, to some extent, handle watershed and flood complexity, they cannot fully describe total complexity of watersheds and their hydrological phenomena [21–23]. They also require field work and a huge bulk of data as well as prohibitive computational costs and parameter estimation [12,14,24]. These shortcomings of classic hydrological models are required to be overcome by new approaches. Tehrany et al. [2] and Bui et al. [25] reported that in recent years, due to aforementioned drawbacks, GIS with data-driven and data mining techniques have offered new insights into natural disaster prediction which can be brought to bear on modeling multi-dimensional floods, an issue which little is known about.

A literature review reveals that all studies in the field of flood hazard modeling can be classified into two main groups including geological-geomorphological and hydrological-hydraulic methods. The first group is based on the field surveys and monitoring of evidence of overflows using remote sensing data [13]. The flood in the second group is simulated and mapped based on the peak flows for specific events or for return periods, thereby obtaining the extent of the water surface [18]. A need to huge amount of data which are not readily available in developing countries and also a need to a great deal of time for model calibration in ungauged hydrometric stations are the weaknesses of the hydrological-hydraulic methods. The generated results by geological-geomorphological methods are also more reasonable than the 1D hydrological-hydraulic methods when river channel is changeable over time and has a high erosive potential [19]. In this regard, some studies have been conducted on flood modeling using hydrological-hydraulic methods including Horritt et al. [21], Di Baldassarre et al. [22], Grimaldi et al. [23], Nguyen et al. [20], and Tyrna et al. [26].

However, many studies have recently been conducted to identify natural hazard prone areas around the world [9,27–31] and to FSM using geological–geomorphological methods such as frequency ratio (FR) [32], weights-of-evidence (WOE) [28], logistic regression (LR) [33], analytical hierarchy process (AHP) [34], artificial neural network (ANN) [35], decision tree (DT) [36], adaptive neuro-fuzzy inference system (ANFIS) [25], ANFIS-genetic algorithm (GA) [37], bagging–LMT hybrid model and random forest [38], and ANFIS-particle swarm optimization (PSO) [39]. Application of machine learning methods in flood studies has been shown by many researchers [40]; however, no general agreement has yet been reached on the selection of the best model for any natural hazard assessment, such as flood susceptibility. Thus, new models are needed and should be tested. Tehrany et al. 2014 [28] have claimed that machine learning is the main source of methods for data driven modeling which can be applied for flooding modeling. Because floods are complex, it is difficult to model them [41], but because data mining and artificial intelligence models have a non-linear structure, they are more proper than other methods. Artificial neural networks have been widely used for natural hazard assessment among other machine learning tools because of their computational efficiency [25,42,43]; however, the modeling may face with errors in some cases due to their poor prediction [25]. Therefore, to bring the disadvantages of the ANN model under control; because ANFIS model high accuracy—which is a hybrid of ANN and fuzzy logic—it has been proposed by some researchers [44–47]. The ANFIS model has a better performance than the two individual models [48,49], but it has some limitations due to its weakness to find the best weight parameters which heavily influence its prediction performance [25] and it is better that these weights searched and determined by using optimization of soft computing techniques. However, there are some optimization methods that have different structures and function distributions to find the weighs of parameters. Additionally, the results of each optimization algorithm are different in each study area due to change in the geo-environmental factors. Therefore, detecting new hybrid algorithms to find the best weights and accessing the reasonable results is a crucial issue in the flood modeling process. The main differences between the current study and other studies in the field of flood susceptibility assessment is that this study uses new hybrid algorithms including ANFIS-CA, ANFIS-BA, and ANFIS-IWO for spatial prediction of floods in the Haraz watershed in the northern part of Iran. Although some optimization and machine learning algorithms have been applied for flood modeling over the world, these optimization methods have not been before explored for flood assessment.

## 2. Study Area

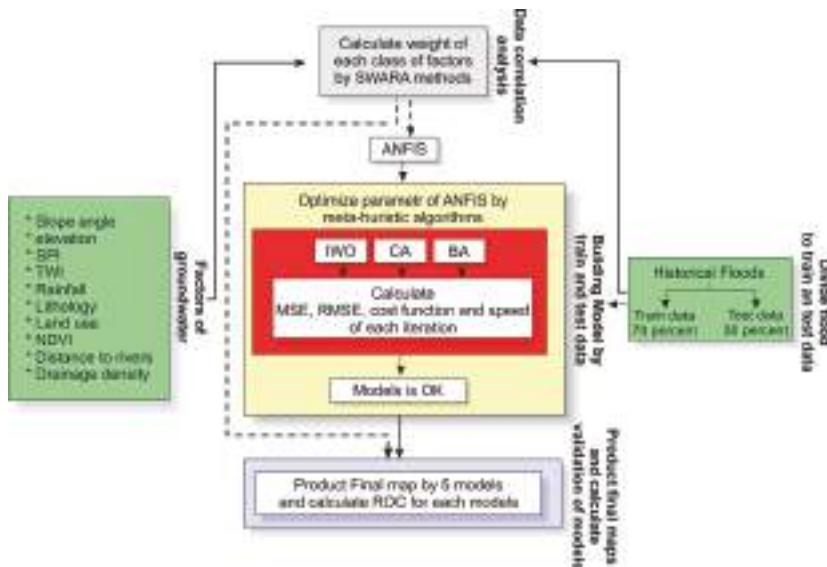
The Haraz watershed is one of the most flood-influenced basins in the Mazandaran province, which was selected for pilot study area. The watershed location is at the south part of Sari a capital city of Mazandaran Province, Iran (Figure 1). It lies between longitudes of  $51^{\circ}43'$  and  $52^{\circ}36'$  E, and the latitudes of  $35^{\circ}45'$  and  $36^{\circ}22'$  N, covering an area of about  $4014 \text{ km}^2$ . The altitude varies from 300 to 5595 m. The climate of the study area based on De Martonne climatic classification system is very humid and its average annual rainfall is 430 mm. The entire study region is high land and mountainous area, for which slopes of  $0\text{--}6^{\circ}$  cover only 5% and ground slope varies from  $0\text{--}66^{\circ}$ . Rangelands cover 92% of the study area, and—geologically—the area is predominantly occupied by Jurassic formations.



**Figure 1.** Flood inventory map and location of the Haraz watershed on Iran map.

### 3. Data Preparation and Analysis

The methodology of the present study is shown in Figure 2, including the following steps.



**Figure 2.** Methodological flowchart used in this study for FSM in the Haraz watershed.

#### 3.1. Flash Flood Inventory

An inventory map is indispensable for future spatial prediction of any natural hazard assessment [50] considering single or multiple events in a specific region for recent and past events [51]. Therefore, the first step in any natural hazard susceptibility assessment is the preparation of the inventory map containing historical records [52]. In the current research, flood inventory map with 201 flood locations was generated using flood historical data from 1995 to 2015 that was finally checked during an extensive field survey.

### 3.2. Dataset Collection for Spatial Modeling

In any natural hazard prediction, spatial relationship between hazard occurrence and conditioning factors should be analyzed [53]. In this research, 10 flood conditioning factors were based on the literature review, data availability, and characteristics of the study area, selected for flooding assessment—including ground slope, altitude, curvature, SPI, TWI, land-use, rainfall, river density, distance to river together with lithology—in a raster format with spatial resolution of 30 m in Environmental Systems Research Institute (ESRI) ArcGIS 10.2 [11,54].

It is very likely for one factor to have a high impact on flooding in a specific watershed while it may not show any influence in another watershed [35]. Since topographical factors are highly significant to identifying flood prone areas and also have a direct impact on the results of modeling [55], a digital elevation model (DEM) of the study area was extracted from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Global DEM with a  $30 \times 30$  m grid size. Five geomorphic factors—such as ground slope, altitude, curvature, SPI, and TWI—were then constructed from DEM. When the ground slope increases, runoff infiltration duration decreases as well and flow velocity increases. Therefore, high volume of runoff enters the river and as result causes floods [2]. The ground slope map was constructed with eight categories [29]: 0–0.5, 0.5–2, 2–5, 5–8, 8–13, 13–20, 20–30, and larger than  $30^\circ$  (Figure 3a). According to previous studies [9,10,29,32], altitude has been considered as one of the most effective factors, and since water flows from higher to lower altitudes in mountainous areas, areas of watershed located in lower altitudes have a higher potential for flood occurrence [18]. The altitude map was prepared with nine classes (Figure 3b): 328–350, 350–400, 400–450, 450–500, 500–1000, 1000–2000, 2000–3000, 3000–4000, and  $> 4000$  m. Flat and concave areas have a higher potential for flooding [2,28,32]. Curvature of the study area was constructed in three categories (Figure 3c):  $<-0.1$ ,  $-0.1\text{--}0.1$ , and  $>0.1$ , namely concave, flat, and convex, respectively.

Hydrological conditioning factors, such as SPI and TWI, affect the spatial variation of flood occurrence. SPI is the erosive power of overland flow [56] and TWI is a topographic index developed by Kirkby and Beven [57] and Beven et al. [58]. TWI indicates water accumulation in a watershed. As the ground slope and catchment area increase, the amount of SPI would increase. The SPI map was constructed with nine categories (Figure 3e): 0–80, 80–400, 400–800, 800–2000, 2000–3000, and  $>3000$  (all of them should be 1000). The TWI map was constructed with six classes (Figure 3d): 1.9–3.94, 3.95–4.47, 4.48–5.03, 5.04–5.71, 5.72–6.96, and 6.97–11.53.

River density and distance to river play notable roles in extend and magnitude of flood occurrence [59]. River network used to prepare river density and distance to river maps. River density was calculated by dividing the river length (m) by the basin area ( $\text{km}^2$ ) [48] and grouped into six categories, including (Figure 3e): 0–0.401, 0.401–1.17, 1.92–2.67, 2.67–3.66, and 3.66–7.3  $\text{km}/\text{km}^2$ . Field survey revealed that there are many flood occurrences adjacent to rivers, and the more the distance to river, the lower the probability of flood occurrence. The distance to river map was constructed using river and multiple ring buffer command in ArcGIS 10.2 and divided into eight classes (Figure 3h): 0–50, 50–100, 100–150, 150–200, 200–400, 400–700, 700–1000, and  $> 1000$  m. A lithology map of the study area with 1:100,000 scale showed six groups of formations including: Teryas, Quaternary, Permain, Cretaceous, Jurassic, and Tertiary (Figure 3i). Land-use type is considered as a conditioning factor that has a significant role in flooding [9]. The areas with higher vegetation density, such as forest regions, can control surface runoff and infiltrate the water; therefore, there is negative spatial relationship between vegetation density and flood occurrence [11]. Land-use map was generated from Landsat 8 Operational Land Imager (OLI) imagery for 2013 in Environment for Visualizing Images (ENVI) 5.1 software (The Board of Trustees of the University of Illinois, Illinois, IL, USA) and classified into seven classes—namely grassland (rangeland), bare land, forest, garden, farming land, residential, and water body—using neural network algorithm and supervised classification (Figure 3j). Rainfall is the most prominent conditioning factor for flood occurrence. About 20 years, from 1991 to 2011, meteorological data was used in order to prepare rainfall maps and then classified into six classes (Figure 3g): 183–333, 334–379, 380–409, 410–448, 449–535, and 536–741 mm using a simple kriging method.

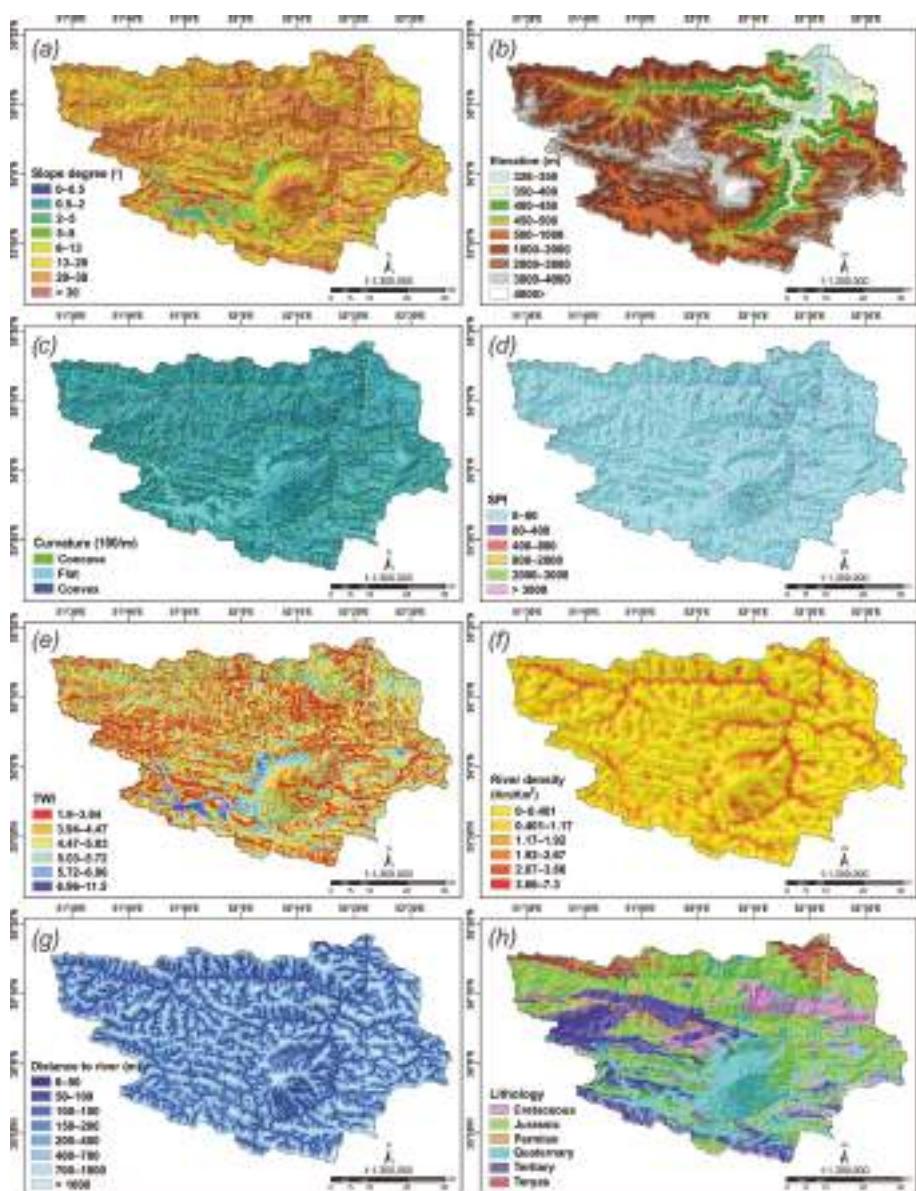
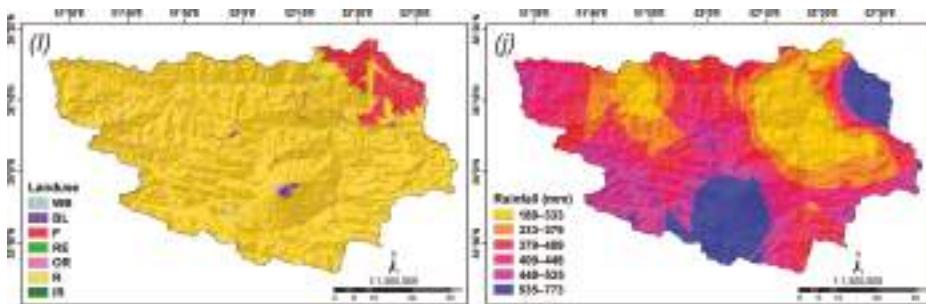


Figure 3. Cont.



**Figure 3.** Flood conditioning factor maps in the study area: slope degree (a), altitude (b), curvature (c), SPI (d), TWI (e), river density (f), distance to river (g), lithology (h), land-use (i), and rainfall (j).

### 3.3. Preparation of Training and Testing Dataset

It should be noted that FSM is considered to have a binary classification in which flood index is divided into two classes, including flood or non-flood; therefore, 201 non-flood locations were first selected by Google Earth on hilly and mountainous regions that are not inundated during flood events [9,10]. To validate model capabilities, the inventory dataset should be divided into two groups [60], one for model construction (training dataset) and the other for model validation (testing dataset). For building the training dataset, 70% of both flood and non-flood locations were randomly selected (141 locations) and then combined together, afterwards to build the testing dataset, 30% of the remaining (60 locations) were incorporated together. Both of training and testing datasets transformed to raster format and overlaid with ten flood conditioning factors at the end. For both datasets, a value of 1 was assigned to flood pixels and a value of 0 to non-flood pixels.

### 3.4. Analysis of Spatial Correlation

SWARA method was used to display the spatial relationship between flood occurrence locations together with each conditioning factors. The SWARA method is one of the new multiple-criteria decision-making (MCDM) techniques developed in 2010 [61]. In this technique, the highest rank is apportioned to the most valuable criteria and the lowest rank is allocated to the least valuable criterion and finally, the mean values of ranks are taken into account to obtain the overall ranks [62]. The advantage of this method is that it enables the expert's opinion in relation to the accuracy of weighting to be assessed in model performance [61]. Another advantage of this method in comparison to other MCDM techniques is that experts can converse and work together, which makes for more accurate results. The SWARA method view point is different from other MCDM methods such as AHP, Analytic network process (ANP), etc. [63]; however, this method leads policy makers to make better decisions according to their goals. The Step-wise weight assessment ratio analysis (SWARA) method was used only for determining the initial weights as input for modeling. In this study, we invited some hydrology experts to prioritize and rank the order of sub-classes of each conditioning factor.

According to Keršuliene et al. [61], the importance relativity of the mean value,  $S_j$  was determined by

$$S_j = \frac{\sum_i^n A_i}{n} \quad (1)$$

where  $n$  stands for the number of experts,  $i$  is counter in sigma,  $A_i$  shows the offered ranks by the experts for each factor, and  $j$  represents the number of the factor.

Afterwards, the coefficient  $K_j$  is determined by

$$K_j = \begin{cases} 1 & j = 1 \\ \frac{S_j + 1}{n} & j > 1 \end{cases} \quad (2)$$

The recalculated weight  $Q_j$  is calculated as

$$Q_j = \frac{Q_{j-1}}{K_j} \quad (3)$$

The comparative weights of evaluation criteria are verbalized as

$$W_j = \frac{Q_j}{\sum_{j=1}^m Q_j} \quad (4)$$

where  $W_j$  presents the relative weight of the  $j$ th criterion, and  $m$  shows the total number of criteria.

### 3.5. Flood Spatial Prediction Modeling

In the present study, three new ensemble models—ANFIS-CA, ANFIS-BA, and ANFIS-IWO—were selected to determine the most susceptible flood areas as well as their comparison.

#### 3.5.1. Adaptive Neuro-Fuzzy Inference System

Takagi and Sugeno [64] presented ANFIS that was obtained from ANN and fuzzy logic [65] by catching the advantages of both in one framework. The ANN's learning capability is automatic. However, this model cannot describe how it acquires the output from decision making. On the contrary, the fuzzy logic can produce output out of fuzzy logic decision, but it does not have the ability to automate learnings [66]. An ANFIS, derived from nature, generates input and output data pairs, so it has been successfully used in diverse fields at solving nonlinear issues and indicating problems [67]. The architecture of ANFIS training is shown in Figure 4. The ANFIS is a feed forward neural network with multi-layer structure. In general, the layers of ANFIS are constructed from six layers with the following function:

Layer 1 (input layer): Layer 1 is the layer of input with the amount of  $x$  and  $y$  passed to the number of neurons in the next layer.

Layer 2 (fuzzification layer): Every node  $i$  in Layer 2, also called the 'fuzzification layer', consists of an adaptive node (square node) which has a node function  $A_i$  or  $B_{i-2}$  as the linguistic label related to the input node  $i$ . Therefore, in Layer 2, fuzzy membership function, is computed which determines 'full', 'partial', or 'none' membership levels. The output function is calculated according to the equations

$$Q_{1,i} = \mu_{A_i}(x_1) \quad \text{for } i = 1, 2 \quad (5)$$

$$Q_{1,i} = \mu_{B_{i-2}}(x_2) \quad \text{for } i = 1, 2 \quad (6)$$

Several membership functions for fuzzifying used many research such as triangular, trapezoidal, Gaussian, and Bell functions. In this research, we used the Bell function such that  $\mu_{A_i}(x_1)$  is given by

$$\mu_{A_i}(x_1) = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^{2b_i}} \quad (7)$$

where  $a_i$ ,  $b_i$ , and  $c_i$  are the parameters of the Bell function that are so-called premise parameters [65,68].

Layer 3 (antecedent layer): All nodes in this layer are fixed nodes labeled as  $\pi$ . In this layer, the aim is to calculate the firing strength for each rule, called  $w_i$  and the following equation computes the outputs

$$Q_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \quad \text{for } i = 1, 2 \quad (8)$$

Layer 4 (strength normalization layer): All nodes in this layer are fixed nodes (square nodes) labeled  $N$ . Each node calculates the individual rules firing strength ratio to the total number of all

rules firing strengths. Output is called the normalized firing strength, determined for each node by the equation

$$Q_{3,i} = \frac{w_i}{\sum w_i} = \frac{w_i}{w_1 + w_2} = w_i \quad \text{for } i = 1, 2 \quad (9)$$

Layer 5 (consequent layer): This layer is the adaptive layer for every node which is known as the defuzzification layer. Every node is an adaptive square node with the node function. It can be defined as

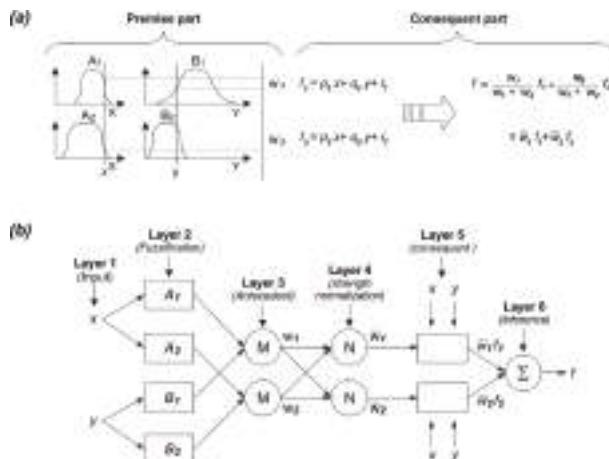
$$Q_{4,1} = w_i \cdot f_i = w_i \cdot (p_i x + q_i y + r_i) \quad \text{for } i = 1, 2 \quad (10)$$

where  $p_i$ ,  $q_i$ , and  $r_i$  are the consequent parameters of function fuzzy inference system ( $f_i$ ).

Layer 6 (inference layer): This layer is a single node which calculates the total number of all received signals from the defuzzification layer in order to generate the overall output shown by circle and labeled  $\Sigma$  and  $f_{out}$  is final output. It can be described as

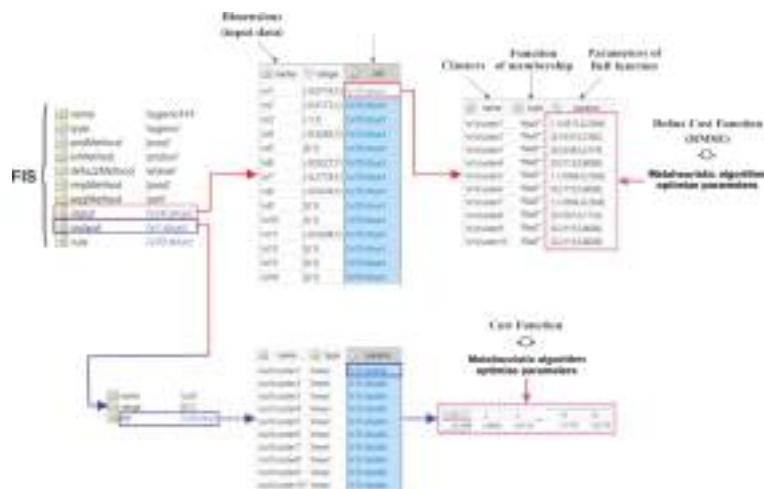
$$Q_{5,1} = \sum w_i \cdot f_i = \frac{\sum w_i \cdot f_i}{\sum w_i} = f_{out} \quad (11)$$

Several researches have recently optimized the parameters of fuzzy membership in ANFIS by using metaheuristic algorithms which has improved the results [69,70]. In this study, cultural algorithms (CA), bees algorithm (BA), and invasive weed optimization algorithm (IWO) were used for optimizing premise parameters of bell function.



**Figure 4.** General ANFIS architecture of first order Takagi–Sugeno fuzzy model [65]: (a) Multi-layer perception fuzzy reasoning; (b) equivalent ANFIS structure.

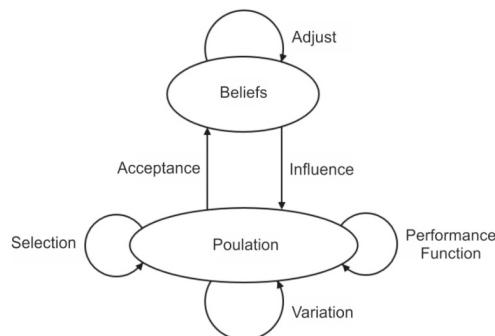
Hybrid models can find the relationships between the SWARA values of each conditioning factor through training dataset. In the modeling process, the optimization algorithms have been performed in three steps (Figure 5) including (i) dividing dataset into two datasets in a 70/30% ratio for model building (training dataset) and model validation (testing dataset), respectively; (ii) construction of bell function in an ANFIS as a membership function and determination of the clusters were performed. Each cluster follows the bell function, that each function parameter of bell function has been optimized by cultural algorithm (CA), bees algorithm (BA), invasive weed optimization algorithm (IWO). We defined the cost function (RSME) and utilize training dataset for optimizing parameters of the bell function of ANFIS; iii) the prediction power of each model has been calculated by the testing dataset.



**Figure 5.** Flowchart of modelling process in this study.

### 3.5.2. Cultural Algorithm

Cultural algorithm (CA) is an evolutionary algorithm introduced by Reynolds [71]. CA expresses an ideal framework of various theories of social evolution with the concept of collective intelligence developed in the 19th century. CA is a computational model of cultural evolution in solving optimization problems that need a vast amount of domain knowledge to steer the collective decisions of individuals in the population. CA has been applied to problems by extensive data, numerous domain limitations, many objectives, and multiple agents in a vast distributed social network. Concluded from social structures, CA compounds evolutionary systems and agents using multiple knowledge sources for the evolution process. Cultural algorithms include two major parts: the space of population and the atmosphere of belief [72]. These two spaces are connected via a communication protocol describing how to link both spaces with the rules for people who believe in their own experiences. These interactions are shown in Figure 6. The population space can include any population-based computational models, like genetic algorithms and evolutionary programming [73]. The belief space supports the information reservoir which every one of their experiences is for other individuals to learn them indirectly.



**Figure 6.** Spaces of a cultural algorithm [74].

### 3.5.3. Bees Algorithm

Another heuristic algorithm is the bees algorithm, which is bee swarm-based. Originally, this algorithm introduced by Pham et al. [75], which is inspired by the foraging behavior of a group of bees when they try to find food resources around their hives [75]. First, the scout bees are distributed uniformly and randomly in different directions to be able to find flower patches in each spot. After identifying the flower patches, the scout bees go back to the hive and start a special dance which is known as the ‘waggle dance’ and it is used for communicating with other group members to share information regarding the flower patches they have already identified. This information may include the direction, the distance to the hive, as well as the amount of nectar existing in the flower patches. The accumulated information shared by these scout bees helps the hive or, in other words, the colony, have a proper assessment of all flower patches available. After accomplishing this phase, the scout bees escort another kind of bees known as the ‘recruit bees’ to go to the discovered flower patches. Different quantities of recruit bees are assigned to each scout bee based on the distance of flower patches as well as the amount of available nectar. In other words, if one flower patch has a higher quality than another one, more recruit bees follow them. Afterwards, the recruit bees constantly assess the flower patch quality while doing the harvest process until the flower patch quality declines. Then, they will leave the flower patch immediately. However, if patch has a steady and satisfactory quality, it will be announced via another waggle dance.

Figure 7 shows the flowchart for bee algorithm. First of all, to implement this algorithm,  $n$  numbers of bees are uniformly distributed in a random manner throughout the search space. The algorithm then starts assessing the fitness for each determined site by the scout bees until the most proper—or optimized—bee is chosen as the ‘elite bee’. The sites of these bees are selected from vicinity of searches’ area and the algorithm explores in the selected sites bees in order to find the best bees at the time when numbers of bees are at its high level. For each site, only the most appropriate bee is being chosen to survive for the next generation. The remaining bees are randomly assigned around the scouting area for new potential solutions. The phase continues so long as the algorithm reaches convergence.



**Figure 7.** Flowchart of the BA for flood susceptibility mapping in Haraz watershed [76].

### 3.5.4. Invasive Weed Optimization Algorithm

Based on the method which originally developed by Mehrabian and Lucas [77] to optimize growth's place of weeds and their reproduction, IWO algorithm is another meta-heuristic algorithm which has been used to imitate the colonizing weeds' behavior. Among the specifications of this algorithm, its simple form, few input parameters, strong robustness, as well as ease of understanding have made it popular for application in problematic nonlinear optimization [78–80]. Furthermore, the IWO algorithm has more optimized solutions for problems in comparison with other algorithms such as particle swarm optimization (PSO) and shuffled frog leaping algorithm (SFLA) and in some cases even it has a better performance than the stated algorithms [78]. The IWO algorithm has five components as follows:

#### (1) Initialization

It involves a random distribution of restricted weeds in the searching space with  $d$  dimension that it is number of factors of flood, which in fact is the earliest population of solutions.

#### (2) Reproduction

During the growing period, weeds are allowed to reproduce a specific number of seeds according to their fitness. As a matter of fact, numbers of reproduced seeds or their  $S_{\min}$  starts from worth fitness and increases to reach  $S_{\max}$  for the weeds with the best fitness, as shown in Figure 8.

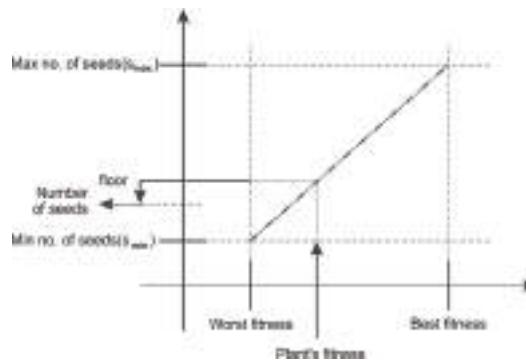


Figure 8. Procedure of seed reproduction at weeds' colony [77].

#### (3) Spatial Dispersal

Reproduced seeds are being distributed by chance all over the searching region. So that they can be located close to their family with the average value which is equal to zero and has non-identical variances. Additionally, in every iteration, the standard deviation (SD) decreases from  $\sigma_{\min}$  to  $\sigma_{\max}$  and is calculated with via non-linear equation which is

$$\sigma_{\text{iter}} = \frac{(\text{iter}_{\max} - \text{iter})^n}{(\text{iter}_{\max})^n} (\sigma_{\min} - \sigma_{\max}) + \sigma_{\max} \quad (12)$$

where  $\text{iter}_{\max}$  is the number of last iteration,  $\sigma_{\text{iter}}$  is the corresponding iteration's SD,  $n$  is the nonlinear modulation index between 2 and 3,  $\sigma_{\max}$  is maximum value's SD and  $\sigma_{\min}$  is minimum value's SD [81].

#### (4) Competitive Exclusion

All weeds together with their seeds are combined to create the population for the next generation. If the population surpasses a definite maximum, the weeds with less fitness will be eradicated.

Reproduction and competition pave the way for the reproduction of the fittest weeds. Therefore, if they produce fitter seeds, seeds can remain alive in the competition.

#### (5) Termination Condition

Steps 2 to 4 are repeated when the iteration reaches to the maximum defined amount and weeds have the maximum fitness that are closest to optimal solution.

#### 3.5.5. Performance Assessment

Quantitative approaches to determine the accuracy of the models are different between observed and estimated values, defined as a forecasting error. In the current study, the capability of models for flood prediction was evaluated using a statistical criterion namely root-mean-squared error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (O_i - E_i)^2}{N}} \quad (13)$$

where  $O_i$  and  $E_i$  are observation and prediction of flood probability values, respectively, in training and testing datasets, and  $N$  is all samples.

#### 3.6. Model Validation and Comparisons

All in all, forecasting capability of flood spatial modeling was analyzed for training and testing datasets [11] using ROC and AUROC, as a standard useful technique to evaluate the prediction capability of models [82,83], the ROC curve is a graph with specificity on the x-axis and sensitivity on the y-axis. Specificity is the number of incorrectly classified floods per total predicted non-floods while sensitivity is the number of correctly classified floods per total predicted floods [84]. The higher the AUROC value is, the better the prediction capability of models will be better [85]. The AUROC can be formulated as

$$\text{AUROC} = \frac{\sum \text{TP} + \sum \text{TN}}{\text{P} + \text{N}} \quad (14)$$

where TP and TN are the number of floods that correctly classified as floods and non-floods, respectively. P and N are the number of total pixels which defines as floods and non-floods, respectively [38]. Results of model performance on training data (success rate) shows a degree of fit of a flood model with the training dataset, indicating how suitable the built model is for flood susceptibility evaluation; therefore, this is not an appropriate method to show the capability of model prediction [51,86,87]. Performance of model using testing/validating dataset (prediction rate) shows how good a model is; thus, this approach should be used for evaluation of model prediction capability. In this research, both success and prediction rates were performed using the training and testing datasets with flood susceptibility index.

#### 3.7. Inferential Statistics

##### 3.7.1. Freidman Test

In the present research, a non-parametric test—namely the Freidman [88] test which is one of the most credible tests for multiple comparisons [89]—was used to find significant differences between models. This test has ranking for each row by considering the rank values of each columns. The null hypothesis ( $H_0$ ) for the current research shows that there is no difference between prediction capabilities of flood models. If the amount of the p-value (significance) is smaller than the significance level ( $\alpha = 0.05$ ), then the null hypothesis is rejected. The biggest weakness of this technique is that it only shows whether there is difference among the models performance or not, and it does not have any capability to display pairwise comparisons among the models.

### 3.7.2. Wilcoxon Test

To overcome Friedman's test weakness, another non-parametric test—Wilcoxon test—was used. This test is used when the aim is to compare two related samples, matched samples, or paired data. The Wilcoxon test prepares pairwise comparisons between all performed flood susceptibility models. The null hypothesis for Wilcoxon test is similar to Friedman's test. The *p*-value and z-score were applied to assess the differences between flood susceptibility models.

## 4. Results

### 4.1. Spatial Relationship between Flood Occurrence and Conditioning Factors

The spatial correlations between flood occurrence and conditioning factors were evaluated, as shown in Table 1. The highest value of SWARA belonged to the first class of 0–0.5 (0.4); therefore, the steeper the ground slope, the lower the flood occurrence probability. The SWARA values decrease when elevation increases and the lowest elevation of 328–350 m had the highest impact (0.63) on the flood occurrence. Generally, the lowest and highest elevations had the highest and lowest influences on flood occurrence, respectively. In the case of curvature, the concave landscape had the highest influence on flooding (0.46), followed by flat (0.43) and then convex (0.11). For SPI, the highest value belonged to class of 2000–3000 (0.32) and values decreased by SPI reducing. The TWI value had a direct impact on flood occurrences events; the greater the TWI, the higher the flood occurrence probability. For the present study, the highest (0.08) and lowest (0) SWARA values belonged to the highest (6.9–11.5) and lowest (1.9–3.9) TWI values. For river density, the class of 2.67–3.66 and 3.66–7.3 showed the highest probability (0.37) and the class of 0–0.4 had the lowest probability (0) of flooding. Results revealed that the more the river density, the higher the flooding probability. The SWARA values showed a decreasing trend when the distance to rivers increased, as the highest and lowest SWARA values belonged to the distance of 0–50 m (0.59) and more than 700 m (0), respectively. According to [2,9,10,28], the most prone areas to flood occurrence were the areas with the lowest elevation, lowest ground slope, flat area, and that were closest to rivers. In the case of lithology, Teryas formations had the highest impact on flooding (0.31), followed by Quaternary (0.21), Permian (0.21), Cretaceous (0.15), Jurassic (0.07), and Tertiary (0.06) formations. Results showed that the land use of water bodies had the highest influence on flooding (0.75), followed by residential area (0.15), garden (0.06), forestlands (0.02), grasslands (0.01), farmlands (0), and barren lands (0). The lowest amount of rainfall (188–333 mm) had the highest impact (0.4) on flooding. In the study area, the more the rainfall, the lower the flooding probability, due to the fact that—in mountainous areas—rainfall would increase with elevation increase; however, flooding occurs at lower elevations.

**Table 1.** Spatial relationship between flood-conditioning factors and flooding occurrences locations by SWARA method

Sub-Factor	Class	Comparative Importance of $K_j$ Average Value	Coefficient $K_j = S_j + 1$	$w_j = (Q_j - 1)/k_j$	Weight $w_j/\sum w_j$
<b>Slope</b>	0–0.5		1.00	1.00	0.40
	0.5–2	0.80	1.80	0.56	0.22
	2–5	0.20	1.20	0.46	0.18
	5–8	0.60	1.60	0.29	0.11
	8–13	1.15	2.15	0.13	0.05
	13–20	1.50	2.50	0.05	0.02
	20–30	0.55	1.55	0.01	0.00
<b>Elevation</b>	>30	2.70	3.70	0.01	0.01
	328–350		1.00	1.00	0.63
	350–400	0.35	1.35	0.16	0.10
	400–450	3.70	4.70	0.21	0.13
	450–500	0.55	1.55	0.10	0.06

**Table 1.** Cont.

Sub-Factor	Class	Comparative Importance of $K_j$ Average Value	Coefficient $K_j = S_j + 1$	$w_j = (Qj - 1)/k_j$	Weight $w_j/\sum w_j$
Curvature	500–1000	0.65	1.65	0.06	0.04
	1000–2000	3.95	4.95	0.01	0.01
	2000–3000	0.00	1.00	0.01	0.01
	3000–4000	0.00	1.00	0.01	0.01
	>4000	0.00	1.00	0.01	0.01
SPI	Concave		1.00	1.00	0.46
	Flat	0.05	1.05	0.95	0.43
	Convex	3.00	4.00	0.24	0.11
TWI	0–80	3.70	4.70	0.09	0.03
	80–400	0.70	1.70	0.41	0.13
	400–800	0.30	1.30	0.70	0.22
	800–2000	0.10	1.10	0.91	0.29
	2000–3000		1.00	1.00	0.32
	>3000	3.95	4.95	0.02	0.01
River density	1.9–3.94	0.05	1.05	0.03	0.00
	3.94–4.47	3.50	4.50	0.03	0.00
	4.47–5.03	2.70	3.70	0.15	0.01
	5.03–5.72	0.65	1.65	0.55	0.04
	5.72–6.96	0.10	1.10	0.91	0.07
	6.96–11.5		1.00	1.00	0.08
Distance to river	0–0.401	3.95	4.95	0.01	0.00
	0.401–1.17	3.95	4.95	0.03	0.01
	1.17–1.92	2.50	3.50	0.15	0.06
	1.92–2.67	0.85	1.85	0.54	0.20
	2.67–3.66		1.00	1.00	0.37
	3.66–7.3	0.00	1.00	1.00	0.37
	7.3–10.0		1.00	1.00	0.00
Lithology	Tertiary		1.00	1.00	0.31
	Quaternary	0.50	1.50	0.67	0.21
	Permian	0.00	1.00	0.67	0.21
	Cretaceous	0.40	1.40	0.48	0.15
	Jurassic	1.10	2.10	0.23	0.07
	Tertiary	0.10	1.10	0.21	0.06
Land use	Water bodies		1.00	1.00	0.75
	Residential area	3.90	4.90	0.20	0.15
	Garden	1.55	2.55	0.08	0.06
	Forest land	2.00	3.00	0.03	0.02
	Grassland	0.70	1.70	0.02	0.01
	Farming land	3.95	4.95	0.00	0.00
	Barren land	0.00	1.00	0.00	0.00
Rainfall	188–333		1.00	1.00	0.40
	333–379	0.10	1.10	0.31	0.12
	379–409	1.20	2.20	0.45	0.18
	409–448	0.35	1.35	0.34	0.13
	448–535	0.05	1.05	0.29	0.12
	535–471	1.15	2.15	0.14	0.05

#### 4.2. Model Comparison between the Proposed New ANFIS Ensemble Models

The newly designed three ANFIS ensemble optimization models—namely ANFIS-CA, ANFIS-BA, and ANFIS-IWO—were used with MATLAB R2016 and ArcGIS 10.2. These models were trained, like other smart models, based on a part of data. Therefore, all data of flood and non-flood points were divided into two categories with the ratio of 30 and 70 percent used as training and test data, respectively.

Training and testing datasets were used as fundamental elements of these models. The training dataset for these three hybrid models was applied to find the relationship between SWARA and values of flood (1) and non-flood (0) locations and finally using testing data, the accuracy of the built model was investigated. Accuracy of training and testing is shown in Figure 9a–f. According to Figure 9a,c,e, the RMSE for ANFIS-CA, ANFIS-BA, and ANFIS-IWO in the training was 0.314, 0.274, and 0.067, respectively. Thus, the hybrid model of ANFIS-IWO showed a better performance with training dataset and had a higher degree of fit; however, the best optimized model was the one which predicted the results of the test data with a higher accuracy. The values of RMSE for testing datasets are shown in Figure 9b,d,f. The values of RMSE for ANFIS-CA, ANFIS-BA, and ANFIS-IWO in testing process were 0.449, 0.365, and 0.359, respectively. Therefore, the ANFIS-IWO optimization algorithm indicated a better performance for both training and testing phases, followed by ANFIS-BA and ANFIS-CA.

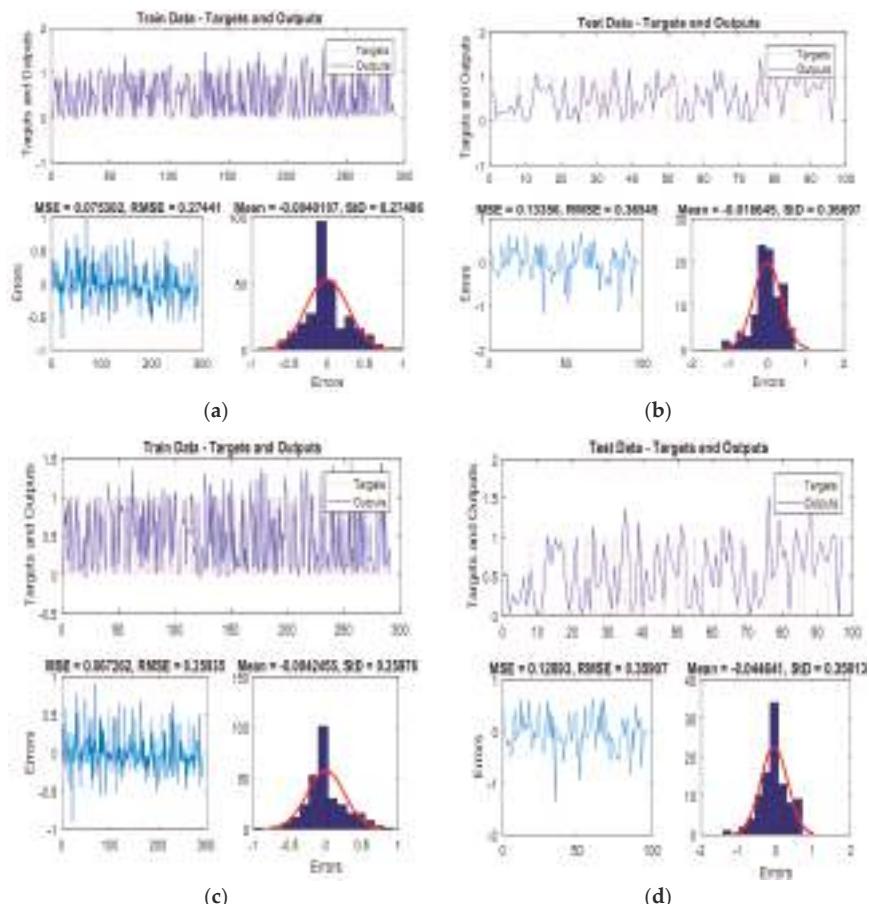
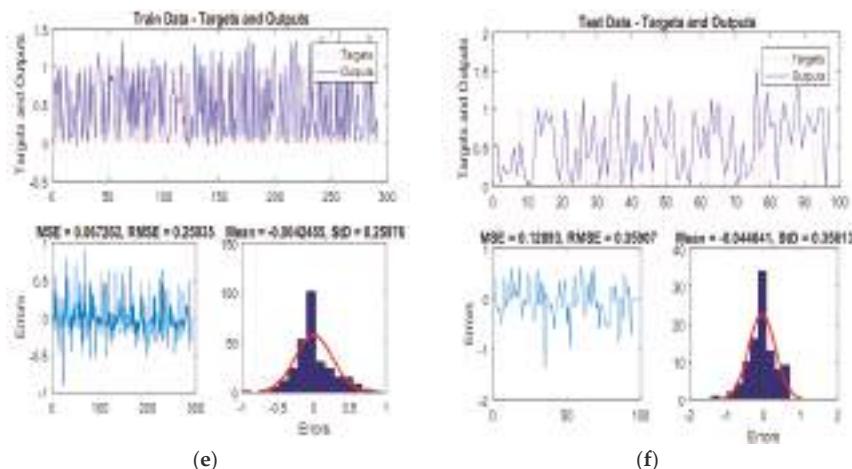
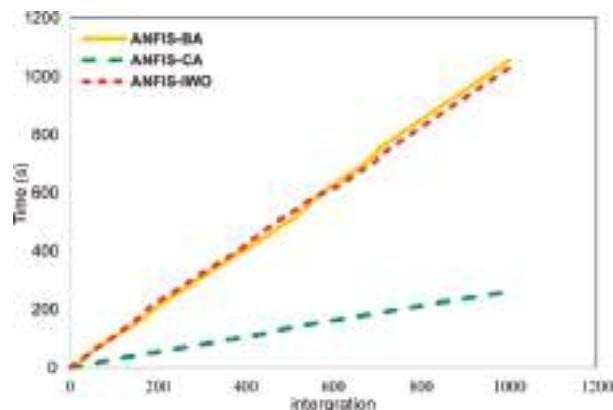


Figure 9. Cont.



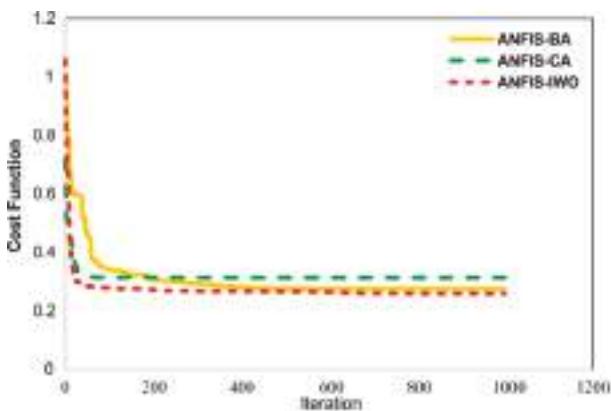
**Figure 9.** RMSE value of training of (a) ANFIS-CA, (c) ANFIS-BA, (e) ANFIS-IWO and for testing data samples (b) ANFIS-CA, (d) ANFIS-BA, and (f) ANFIS-IWO.

In addition, it should be noted that the processing speed of the models is also important nowadays. The processing time of 1000 iterations was then estimated as 1, 120, 260 s and 1100 s for the ANFIS-BA, ANFIS-CA, and ANFIS-IWO hybrid models by coding, respectively. As a result, the processing speed of ANFIS-CA had the least time, and the ANFIS-BA model had the highest process time. On the other hand, we examined the convergence of each model in the training phase (Figure 10).



**Figure 10.** Cumulative curve of speed processing from applied models.

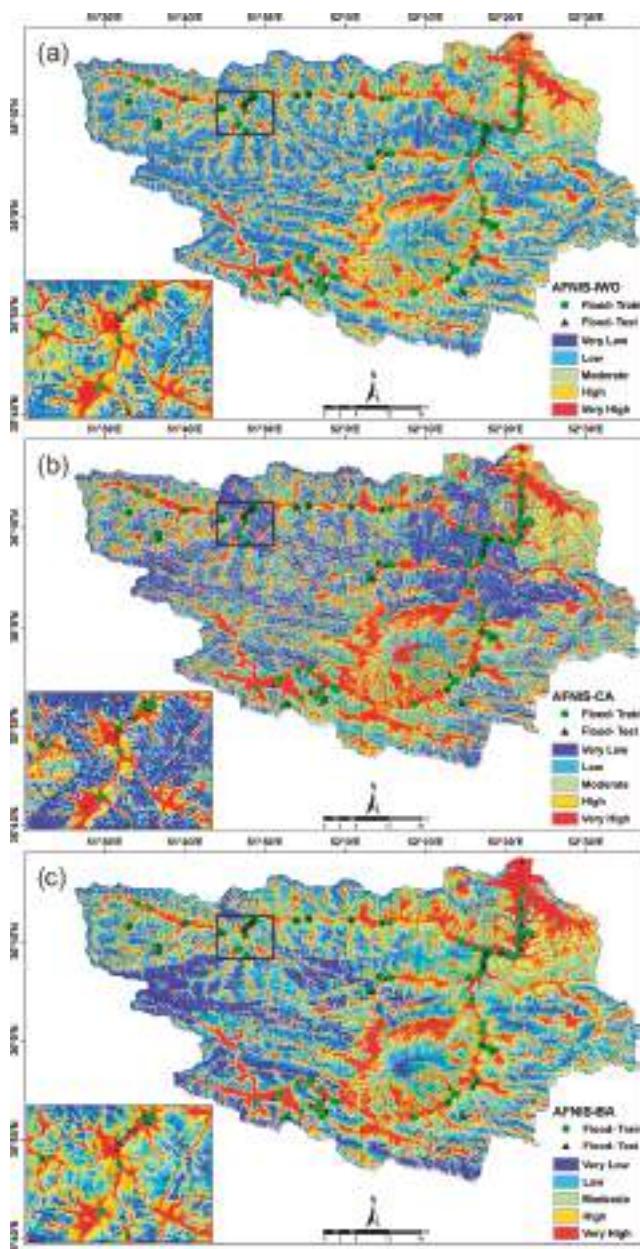
The convergence curve was gained by graphing the calculated function of cost in each iteration of three models (Figure 11). Results showed that cost-function values of the ANFIS-CA model were fixed in the 25th iteration, which indicates an iteration convergence of this model in comparison to other models. On the contrary, ANFIS-BA and ANFIS-IWO models converged in 450th and 625th iterations indicating the slow speed of these models in achieving convergence, respectively.



**Figure 11.** Speed of convergence of applied models.

#### 4.3. Model Configuration and Generating of FSMs Using ANFIS Ensemble Models

As the main aim of flood susceptibility modeling was the reorganization of prone areas with higher probability of flooding; therefore, three optimization models—namely cultural, bees, and invasive weed optimization algorithms—were combined with SWARA-ANFIS to optimize the model for identifying flood prone areas with higher accuracy. ANFIS hybrid models were built using the training dataset and SWARA method values which were standardized between 0 and 1 in MATLAB R2016 software, The MathWorks, Inc, Massachusetts, MA, USA. Then, the constructed models were applied to the entire study area to create flood susceptibility probability (indices) and finally these indices for each pixel size (pixel-by-pixel) was used to create the final FSMs in ArcGIS 10.2 for the Haraz watershed. In the next step, indexes were reclassified into five classes (very low, low, moderate, high, and very high susceptibility) using the quantile method [18]. Three FSMs were then developed for comparative visualization, as shown in Figure 12a–c. Results demonstrated that the areas near rivers with lower slopes and altitudes had higher probabilities of flooding.

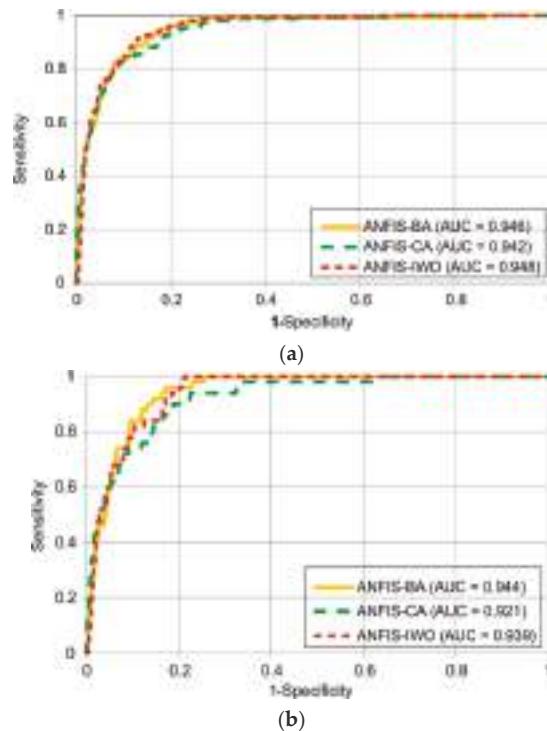


**Figure 12.** FSM using ANFIS-IWO (a), ANFIS-CA (b), and ANFIS-BA (c) for Haraz watershed.

#### 4.4. Validation of Flood Susceptibility Maps

For validation of the three FSMs, both success and prediction rate curves were applied. The ROC plots are shown in Figure 13a,b. For the training phase, the ANFIS-IWO had the highest AUROC (0.948), followed by ANFIS-BA (0.946) and ANFIS-CA (0.942), implying that ANFIS-IWO had a

fitter degree with training dataset. The validation of three obtained maps for the testing dataset revealed that ANFIS-BA (0.944) had a higher flood prediction capability for the Haraz watershed, followed by ANFIS-IWO (0.939) and ANFIS-CA (0.932). According to the relationship between AUROC and prediction capabilities of models, these models showed excellent performances based on the following classification: 0.5–0.6 (poor), 0.6–0.7 (average), 0.7–0.8 (good), 0.8–0.9 (very good), and 0.9–1 (excellent) [25,90].

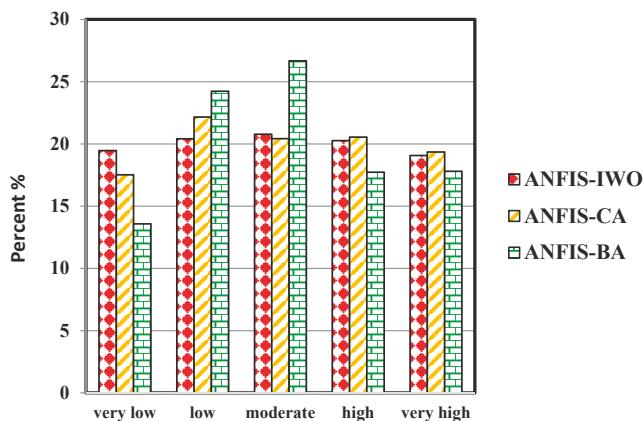


**Figure 13.** Model validation by success rate (a) and prediction rate (b) for three hybrid models.

As the most appropriate model to predict FSM for the Haraz watershed was the ANFIS-BA algorithm, the other two hybrid models had overestimated the results in very low, high, and very high susceptibility classes and underestimated in low and moderate classes (Figure 14). It can be seen that, according to the ANFIS-BA algorithm, the very high susceptibility class covered 17% of the study area, and analysis of flood location revealed that about 63% of total flood locations was located in this class. Overall, the very high and high classes contained about 85% of the total flood locations. The low and moderate classes covered about 50% of the study area according to the ANFIS-BA algorithm.

Results of the Friedman and Wilcoxon tests are shown in Tables 2 and 3. They revealed that since the *p*-value was less than 0.05 (0.00) and chi-square was more than 3.84 (standard value), the null hypothesis was rejected, indicating there were significant differences among the three flood susceptibility models.

The Wilcoxon test was carried out to check the statistical significance of pairwise differences between flood models. Based on the judgment, there were significant differences between ANFIS-CA and ANFIS-BA and ANFIS-CA and ANFIS-IWO; however, there was no significant difference between ANFIS-BA and ANFIS-IWO algorithms.



**Figure 14.** Percentages of different flood susceptibility classes in the Haraz watershed.

**Table 2.** Friedman test achievement

Number	Flood Models	Mean Ranks	Chi-Square	p-Value (Significance)
1	ANFIS-CA	1.68		
2	ANFIS-BA	2.08	16.6	0.00
3	ANFIS-IWO	2.24		

**Table 3.** Results of Wilcoxon signed-rank test

Number	Pairwise Comparison	z-Score	p-Value (Significance)	Judgment
1	ANFIS-CA vs. ANFIS-BA	-3.225	0.001	Yes
2	ANFIS-CA vs. ANFIS-IWO	-3.906	0.000	Yes
3	ANFIS-BA vs. ANFIS-IWO	-1.128	0.259	NO

## 5. Discussion

Flooding is known as the most frequent and destructive natural hazard. The occurrence of floods is increasing worldwide and its human losses and socio-economic damages pose huge pressure on communities. This trend is occurring in northern Iran as well, especially in the Haraz watershed. This huge burden of floods on human societies has brought about geophysicists, hydrologists, water resources engineers, geologists, and geomorphologists to study this phenomenon from different aspects such that its management can be met and its financial damages can be mitigated. One of the most important tasks of these scientists is to identify the areas within a watershed through different tools that are very vulnerable to flood generation. Field work might be preferable; however, it is costly, time-consuming, and hard to conduct. Therefore, a modeling approach is always an alternate tool to direct measurement of flood characteristics. The present study introduces three novel hybrid models—ANFIS-CA, ANFIS-BA, and ANFIS-IWO algorithms—for identifying flood prone areas of the Haraz watershed with a high precision and compares their prediction capabilities. In this study, the SWARA method was adopted to select the most effective factors and the most effective class of each of 10 conditioning factors. Our finding indicated that all factors had significant relationships with floods such that they were all selected for flood susceptibility modeling. Three flood susceptibility maps were generated using three above-mentioned novel hybrid models and they were then classified into five classes.

A comparison between the results of modeling process and five flood susceptibility classes including very low, low, moderate, high, and very high susceptibility classless on the terrain was taken place. Basically, in the very high and high susceptibility classes, all hybrid models covered low lands (flats) and the areas around the rivers. Most of observation floods are occurred due to overbanking the flow. Also, the floodplains over the study area are surrounded by this class of susceptibility. This region located generally in the elevations less than 400 m above sea level, slope angle less than 8% in concave slopes with rainfall more than about 400 mm and the distance less than 200 m from the rivers. The obtained results of modeling process indicated that the ANFIS-CA and ANFIS-IWO have occupied more pixels of very high susceptibility class in comparison to ANFIS-BA algorithm.

In the low and moderate susceptibility class, with increasing the distance from the rivers, the areas where are covered by floods will be decreased. This region generally occupied by slope between 8% and 12%, elevation above sea level between 400 and 2000 m, concave and convex slopes, and the distance from the river between 200 and 700 m. The results of modeling process concluded that the ANFIS-BA hybrid model has covered most areas of the study area in comparison to the ANFIS-CA and ANFIS-IWO hybrid models.

The very low susceptibility class mainly included the mountainous areas and hill slopes which are much far away from the rivers compared to the other susceptibility classes. In term of topography, this zone is usually located in the areas where the slope angle and elevation above-sea level are more than 20% and 2000 m, respectively. The slope shape in this class is mainly concave and convex slopes and the distance from the river is more than 700 m. The modelling process revealed that the ANFIS-BA hybrid model has the least areas over the study area unlike to the ANFIS-CA and ANFIS-IWO hybrid models.

Three ANFIS ensemble optimization approaches—including ANFIS-CA, ANFIS-BA and ANFIS-IWO—were designed to flood modeling of the Haraz watershed for which the entire dataset was divided into two series; 70% of the data for training of models and the rest for testing. Results of model evaluation using RMSE for the best degree of fit in the training phase and the best prediction accuracy in the testing phase revealed that the ANFIS-IWO optimization algorithm indicated a better performance in training and testing phases, followed by ANFIS-BA and ANFIS-CA, while all three models provided acceptable results for flood modeling. However, results of the AUC showed that ANFIS-IWO had the best degree of fit in the training phase which was coincident with the results of RMSE; whereas, in the testing phase, the results of AUC revealed that ANFIS-BA had the best prediction power than other models. The most important disadvantage of RMSE was that it acted based only on error assessments; therefore, better approaches must be found to resolve this weak spot, as the model should be chosen based on its abilities. Treating the ROC and AUC based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN) showed more accurate evaluation of the models than RMSE [39]. According to AUC, the ANFIS-IWO had the fittest degree to the training data; however, the ANFIS-BA had the highest prediction capabilities compare other methods in testing phase. As the testing dataset was not used for modeling, the model evaluation must be performed based on the testing dataset. The challenge here is determining how it is possible for a model to show highest degree of fit in the training phase (ANFIS-IWO) but not to have the highest prediction power in the testing phase. Termeh et al. [39] stated that this situation can take place in the models with the over fitting problem. The BA algorithm used a series of parameters including number of scout bees, best patches, as well as elite patches in the selected patches, amount of employed bees in the elite patches, number of recruited bees in the non-elite best patches, the neighborhood size for each patch, number of iterations, together with difference between first and last iterations value; make it robust. The bees algorithm has local and global searching capabilities that find the best locations. Finally, three designed models were also evaluated for processing speed and convergence criterion. Results indicated that although ANFIS-CA had the lowest prediction power than other models but acquired the lowest processing speed and the highest speed of convergence.

Some studies focused on spatial prediction of floods using machine learning and optimization algorithms over the world. In this case, Chapi et al. [38] introduced a new machine learning algorithm, namely bagging-logistic model tree (BA-LMT) for flood modelling. They concluded that the proposed ensemble model had outperformed the LMT, the logistic regression, the Bayesian logistic regression, and the random forest algorithms. Khosravi et al. [36] compared some soft computing benchmark machine learning algorithms—such as logistic model trees (LMT), reduced error pruning trees (REPT), naïve Bayes trees (NBT), and alternating decision trees (ADT)—for flash flood susceptibility mapping. They revealed that the ADT model had the highest prediction capability, followed by the NBT, LMT, and REPT algorithms, respectively. The results of above mentioned studies indicated the capability of machine learning algorithms in spatial prediction of floods. However, there are some other studies which conducted on optimization as evolutionary algorithms for flood susceptibility assessment such as Bui et al. [25], Hong et al. [24], Ahmadlou et al. [70] and Termeh et al. [39]. Among these studies, Bui et al. [25] introduced a new hybrid model of neural fuzzy inference system and metaheuristic optimization including evolutionary genetic and particle swarm optimization (MONF) for flood mapping. The obtained results were confirmed and compared with the J48 decision tree, the random forest, the multi-layer perceptron neural network, the support vector machine, and the adaptive neuro fuzzy inference system (ANFIS) algorithms. Additionally, Hong et al. [24] concluded that the new hybrid optimization algorithm of ANFIS-differential evolution (ANFIS-DE) outperformed and outclassed of the ANFIS-genetic algorithm (ANFIS-GA). Also, the results of Termeh et al. [39] indicated the highest capability of ANFIS-PSO ensemble model in comparison to ANFIS-DE and ANFIS-GA models. The mentioned studies indicated that the optimization algorithms for flood modeling in the different study areas over the world had more power prediction in comparison to machine learning algorithms. Basically, the results of this study which illustrated the ability of new optimization algorithm—namely bees—for flood susceptibility assessment in the study area are in agreement with other mentioned studies over the world. The bees algorithm is a more robust optimization algorithm in comparison to the CA and IWO algorithms in the study area.

What is pointed out from the above-mentioned studies is that each model has its own advantages and disadvantages. Thus, the new models must be applied and the model with the highest prediction power selected for further and future decisions. Moreover, it is better to state that some of machine learning models have weaknesses; thus, new hybrid models should be continuously introduced to resolve this problem [32]. The current research like other studies has some uncertainties especially in (1) variable factors, such as rainfall, land-use; (2) impact of climate change on flood occurrences; (3) impact of human activities on flood occurrences; and (4) determination of flood inundation map along with flood susceptibility maps. Thus, for future cases, it is recommended that 2-D flood inundation maps of flood-prone areas achieved by the current research should be produced using some commonly hydraulic models such as Hydrologic Engineering Center-River Analysis System (HEC-RAS) software.

## 6. Conclusions

FSM especially in ungauged watersheds has a scientific and practical value in the context of basin-scale water resources management and is a base for hazard and risk mapping. It can be considered as a useful tool for land-use planning, decision making, and flood disaster management. Due to complexity and non-linearity structure of watersheds, floods cannot be modeled by employing classic statistical and physically-based distributed methods; therefore, for the determination of flood prone areas in the Haraz watershed, Iran, with high accuracy, new hybrid artificial intelligence models were used. To fulfill this purpose, 201 flood locations were collected and randomly divided into two groups for training and validation dataset, and 10 flood conditioning factors were then selected. ANFIS weighted by SWARA method was applied to make an initial flood susceptibility model and three optimization models—namely CA, BA, and IWO—were adopted to optimize the models. These optimization techniques find optimal parameters and also decrease the problems of local minimum and are therefore appropriate for the training of artificial intelligence methods and

optimizing their results as well, especially for solving complex problems such as flood prediction. It is obvious that more precise flood susceptibility maps can decrease the cost and damages from flooding. Finally, FSMs were constructed using ANFIS-CA, ANFIS-BA, and ANFIS-IWO and then classified into five categories using the quantile method in order to produce the susceptibility maps and the results of the achieved map represented very low, low, moderate, high, and very high susceptibility areas. Finally, these three new hybrid optimization models were evaluated using ROC and AUROC. Results revealed that ANFIS-IWO algorithm had a better degree of fit with the training dataset but ANFIS-BA had a higher prediction capability in FSM using the testing dataset followed by ANFIS-IWO and ANFIS-CA.

**Author Contributions:** D.T.B., K.K., S.L., H.S., M.P., V.P.S., K.C., A.S., S.P., W.C. and B.B.A contributed equally to the work. K.K. and M.P. collected field data and conducted the flood mapping and analysis. K.K., H.S., M.P., K.C., A.S. and S.P. wrote the manuscript. D.T.B., S.L., V.P.S., W.C. and B.B.A provided critical comments in planning this paper and edited the manuscript. All the authors discussed the results and edited the manuscript.

**Funding:** This research was financial supported by International Partnership Program of Chinese Academy of Sciences (grant no. 115242KYSB20170022), Iran National Science Foundation (INSF) through the research project no. 96004000, and Universiti Teknologi Malaysia (UTM) based on the Research University Grant (QJ130000.2527.17H84).

**Acknowledgments:** We express our thanks to Editor-in-Chief of the Water journal and our two anonymous reviewers. With their comments and suggestions, we were able to significantly improve the quality of our paper.

**Conflicts of Interest:** No potential conflict of interest was reported by the authors.

## References

- Elkhrachy, I. Flash flood hazard mapping using satellite images and GIS tools: A case study of Najran City, Kingdom of Saudi Arabia (KSA). *Egypt. J. Remote Sens. Space. Sci.* **2015**, *18*, 261–278. [[CrossRef](#)]
- Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Flood susceptibility analysis and its verification using a novel ensemble support vector machine and frequency ratio method. *Stoch. Environ. Res. Risk Assess.* **2015**, *29*, 1149–1165. [[CrossRef](#)]
- Youssef, A.M.; Pradhan, B.; Hassan, A.M. Flash flood risk estimation along the St. Katherine road, southern Sinai, Egypt using GIS based morphometry and satellite imagery. *Environ. Earth Sci.* **2011**, *62*, 611–623. [[CrossRef](#)]
- Lee, E.H.; Kim, J.H.; Choo, Y.M.; Jo, D.J. Application of flood nomograph for flood forecasting in urban areas. *Water* **2018**, *10*, 53. [[CrossRef](#)]
- Sarhadi, A.; Soltani, S.; Modarres, R. Probabilistic flood inundation mapping of ungauged rivers: Linking GIS techniques and frequency analysis. *J. Hydrol.* **2012**, *458*, 68–86. [[CrossRef](#)]
- Luu, C.; von Meding, J. A flood risk assessment of quang nam, vietnam using spatial multicriteria decision analysis. *Water* **2018**, *10*, 461. [[CrossRef](#)]
- Dutta, D.; Herath, S. Trend of Floods in Asia and Flood Risk Management with Integrated River Basin Approach. In Proceedings of the 2nd APHW Conference, Singapore, 5–8 July 2004; pp. 55–63.
- Smith, K. *Environmental Hazards: Assessing Risk and Reducing Disaster*; Routledge: London, UK, 2013.
- Khosravi, K.; Nohani, E.; Maroufinia, E.; Pourghasemi, H.R. A GIS-based flood susceptibility assessment and its mapping in iran: A comparison between frequency ratio and weights-of-evidence bivariate statistical models with multi-criteria decision-making technique. *Nat. Hazards* **2016**, *83*, 947–987. [[CrossRef](#)]
- Khosravi, K.; Pourghasemi, H.R.; Chapi, K.; Bahri, M. Flash flood susceptibility analysis and its mapping using different bivariate models in Iran: A comparison between Shannon’s entropy, statistical index, and weighting factor models. *Environ. Monit. Assess.* **2016**, *188*, 656. [[CrossRef](#)] [[PubMed](#)]
- Shafizadeh-Moghadam, H.; Valavi, R.; Shahabi, H.; Chapi, K.; Shirzadi, A. Novel forecasting approaches using combination of machine learning and statistical models for flood susceptibility mapping. *J. Environ. Manag.* **2018**, *217*, 1–11. [[CrossRef](#)] [[PubMed](#)]
- Chapi, K. Monitoring and Modeling of Runoff Generating Areas in a Small Agricultural Watershed. Ph.D. Thesis, University of Guelph, Guelph, Canada, 2010.

13. Chapi, K.; Rudra, R.P.; Ahmed, S.I.; Khan, A.A.; Gharabaghi, B.; Dickinson, W.T.; Goel, P.K. Spatial-temporal dynamics of runoff generation areas in a small agricultural watershed in southern Ontario. *J. Water Resour. Prot.* **2015**, *7*, 14–40. [[CrossRef](#)]
14. Fenicia, F.; Kavetski, D.; Savenije, H.H.; Clark, M.P.; Schoups, G.; Pfister, L.; Freer, J. Catchment properties, function, and conceptual model representation: Is there a correspondence? *Hydrol. Process.* **2014**, *28*, 2451–2467. [[CrossRef](#)]
15. Kisi, O.; Nia, A.M.; Gosheh, M.G.; Tajabadi, M.R.J.; Ahmadi, A. Intermittent streamflow forecasting by using several data driven techniques. *Water Resour. Manag.* **2012**, *26*, 457–474. [[CrossRef](#)]
16. Ganguli, P.; Reddy, M.J. Probabilistic assessment of flood risks using trivariate copulas. *Theor. Appl. Climatol.* **2013**, *111*, 341–360. [[CrossRef](#)]
17. Refsgaard, J.C. Parameterisation, calibration and validation of distributed hydrological models. *J. Hydrol.* **1997**, *198*, 69–97. [[CrossRef](#)]
18. Cea, L.; Bladé, E. A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications. *Water Resour. Res.* **2015**, *51*, 5464–5486. [[CrossRef](#)]
19. Costabile, P.; Costanzo, C.; Macchione, F. A storm event watershed model for surface runoff based on 2D fully dynamic wave equations. *Hydrol. Process.* **2013**, *27*, 554–569. [[CrossRef](#)]
20. Xia, X.; Liang, Q.; Ming, X.; Hou, J. An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations. *Water Resour. Res.* **2017**, *53*, 3730–3759. [[CrossRef](#)]
21. Bellos, V.; Tsakiris, G. A hybrid method for flood simulation in small catchments combining hydrodynamic and hydrological techniques. *J. Hydrol.* **2016**, *540*, 331–339. [[CrossRef](#)]
22. Liang, D.; Özgen, I.; Hinkelmann, R.; Xiao, Y.; Chen, J.M. Shallow water simulation of overland flows in idealised catchments. *Environ. Earth Sci.* **2015**, *74*, 7307–7318. [[CrossRef](#)]
23. Singh, J.; Altinakar, M.S.; Ding, Y. Numerical modeling of rainfall-generated overland flow using nonlinear shallow-water equations. *J. Hydrol. Eng.* **2014**, *20*. [[CrossRef](#)]
24. Rahmati, O.; Naghibi, S.A.; Shahabi, H.; Bui, D.T.; Pradhan, B.; Azareh, A.; Rafiei-Sardooi, E.; Samani, A.N.; Melesse, A.M. Groundwater spring potential modelling: comprising the capability and robustness of three different modeling approaches. *J. Hydrol.* **2018**, *565*, 248–261. [[CrossRef](#)]
25. Bui, D.T.; Pradhan, B.; Nampak, H.; Bui, Q.-T.; Tran, Q.-A.; Nguyen, Q.-P. Hybrid artificial intelligence approach based on neural fuzzy inference model and metaheuristic optimization for flood susceptibility modeling in a high-frequency tropical cyclone area using GIS. *J. Hydrol.* **2016**, *540*, 317–330.
26. Tyrna, B.; Assmann, A.; Fritsch, K.; Johann, G. Large-scale high-resolution pluvial flood hazard mapping using the raster-based hydrodynamic two-dimensional model flood FloodAreaHPC. *J. Flood Risk Manag.* **2018**, *11*, S1024–S1037. [[CrossRef](#)]
27. Al-Abadi, A.M.; Shahid, S.; Al-Ali, A.K. A GIS-based integration of catastrophe theory and analytical hierarchy process for mapping flood susceptibility: A case study of teeb area, southern Iraq. *Environ. Earth Sci.* **2016**, *75*, 1–19. [[CrossRef](#)]
28. Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in GIS. *J. Hydrol.* **2014**, *512*, 332–343. [[CrossRef](#)]
29. Tehrany, M.S.; Pradhan, B.; Mansor, S.; Ahmad, N. Flood susceptibility assessment using GIS-based support vector machine model with different kernel types. *Catena* **2015**, *125*, 91–101. [[CrossRef](#)]
30. Youssef, A.M.; Pradhan, B.; Sefry, S.A. Flash flood susceptibility assessment in Jeddah city (Kingdom of Saudi Arabia) using bivariate and multivariate statistical models. *Environ. Earth Sci.* **2016**, *75*, 12. [[CrossRef](#)]
31. Fotopoulos, F.; Makropoulos, C.; Mimikou, M.A. Validation of satellite rainfall products for operational flood forecasting: The case of the Evros catchment. *Theor. Appl. Climatol.* **2011**, *104*, 403–414. [[CrossRef](#)]
32. Tehrany, M.S.; Pradhan, B.; Jebur, M.N. Spatial prediction of flood susceptible areas using rule based decision tree (DT) and a novel ensemble bivariate and multivariate statistical models in GIS. *J. Hydrol.* **2013**, *504*, 69–79. [[CrossRef](#)]
33. Pradhan, B. Flood susceptible mapping and risk area delineation using logistic regression, GIS and remote sensing. *J. Spatial Hydrol.* **2010**, *9*, 1–18.
34. Gigović, L.; Pamučar, D.; Bajić, Z.; Drobnjak, S. Application of GIS-interval rough AHP methodology for flood hazard mapping in urban areas. *Water* **2017**, *9*, 360. [[CrossRef](#)]

35. Kia, M.B.; Pirasteh, S.; Pradhan, B.; Mahmud, A.R.; Sulaiman, W.N.A.; Moradi, A. An artificial neural network model for flood simulation using GIS: Johor River Basin, Malaysia. *Environ. Earth Sci.* **2012**, *67*, 251–264. [[CrossRef](#)]
36. Khosravi, K.; Pham, B.T.; Chapi, K.; Shirzadi, A.; Shahabi, H.; Revhaug, I.; Prakash, I.; Bui, D.T. A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at Haraz watershed, northern Iran. *Sci. Total Environ.* **2018**, *627*, 744–755. [[CrossRef](#)] [[PubMed](#)]
37. Hong, H.; Panahi, M.; Shirzadi, A.; Ma, T.; Liu, J.; Zhu, A.-X.; Chen, W.; Koulias, I.; Kazakis, N. Flood susceptibility assessment in Hengfeng area coupling adaptive neuro-fuzzy inference system with genetic algorithm and differential evolution. *Sci. Total Environ.* **2017**, *621*, 1124–1141. [[CrossRef](#)] [[PubMed](#)]
38. Chapi, K.; Singh, V.P.; Shirzadi, A.; Shahabi, H.; Bui, D.T.; Pham, B.T.; Khosravi, K. A novel hybrid artificial intelligence approach for flood susceptibility assessment. *Environ. Model. Softw.* **2017**, *95*, 229–245. [[CrossRef](#)]
39. Termeh, S.V.R.; Kornejady, A.; Pourghasemi, H.R.; Keesstra, S. Flood susceptibility mapping using novel ensembles of adaptive neuro fuzzy inference system and metaheuristic algorithms. *Sci. Total Environ.* **2018**, *615*, 438–451. [[CrossRef](#)] [[PubMed](#)]
40. Maier, H.R.; Jain, A.; Dandy, G.C.; Sudheer, K.P. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environ. Model. Softw.* **2010**, *25*, 891–909. [[CrossRef](#)]
41. Dottori, F.; Martina, M.L.V.; Figueiredo, R. A methodology for flood susceptibility and vulnerability analysis in complex flood scenarios. *J. Flood Risk Manag.* **2018**, *11*, S632–S645. [[CrossRef](#)]
42. Ghalkhani, H.; Golian, S.; Saghafian, B.; Farokhnia, A.; Shamseldin, A. Application of surrogate artificial intelligent models for real-time flood routing. *Water Environ. J.* **2013**, *27*, 535–548. [[CrossRef](#)]
43. Rezaeianzadeh, M.; Tabari, H.; Yazdi, A.A.; Isik, S.; Kalin, L. Flood flow forecasting using ANN, ANFIS and regression models. *Neural Comput. Appl.* **2014**, *25*, 25–37. [[CrossRef](#)]
44. Chang, F.-J.; Tsai, M.-J. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *J. Hydrol.* **2016**, *535*, 256–269. [[CrossRef](#)]
45. Güçlü, Y.S.; Şen, Z. Hydrograph estimation with fuzzy chain model. *J. Hydrol.* **2016**, *538*, 587–597. [[CrossRef](#)]
46. Lohani, A.; Kumar, R.; Singh, R. Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. *J. Hydrol.* **2012**, *442*, 23–35. [[CrossRef](#)]
47. Shu, C.; Ouarda, T. Regional flood frequency analysis at ungauged sites using the adaptive neuro-fuzzy inference system. *J. Hydrol.* **2008**, *349*, 31–43. [[CrossRef](#)]
48. Mukerji, A.; Chatterjee, C.; Raghuwanshi, N.S. Flood forecasting using ANN, Neuro-Fuzzy, and Neuro-GA models. *J. Hydrol. Eng.* **2009**, *14*, 647–652. [[CrossRef](#)]
49. Nayak, P.C.; Sudheer, K.P.; Rangan, D.M.; Ramasastri, K.S. Short-term flood forecasting with a neurofuzzy model. *Water Resour. Res.* **2005**, *41*. [[CrossRef](#)]
50. Pham, B.T.; Bui, D.T.; Dholakia, M.; Prakash, I.; Pham, H.V. A comparative study of least square support vector machines and multiclass alternating decision trees for spatial prediction of rainfall-induced landslides in a tropical cyclones area. *Geotech. Geol. Eng.* **2016**, *34*, 1807–1824. [[CrossRef](#)]
51. Tien Bui, D.; Pradhan, B.; Lofman, O.; Revhaug, I. Landslide susceptibility assessment in vietnam using support vector machines, decision tree, and Naive Bayes Models. *Math. Probl. Eng.* **2012**, *2012*, 974638. [[CrossRef](#)]
52. Guzzetti, F.; Mondini, A.C.; Cardinali, M.; Fiorucci, F.; Santangelo, M.; Chang, K.-T. Landslide inventory maps: New tools for an old problem. *Earth Sci. Rev.* **2012**, *112*, 42–66. [[CrossRef](#)]
53. Pham, B.T.; Pradhan, B.; Bui, D.T.; Prakash, I.; Dholakia, M. A comparative study of different machine learning methods for landslide susceptibility assessment: A case study of Uttarakhand area (India). *Environ. Model. Softw.* **2016**, *84*, 240–250. [[CrossRef](#)]
54. Akgun, A. A comparison of landslide susceptibility maps produced by logistic regression, multi-criteria decision, and likelihood ratio methods: A case study at Izmir, turkey. *Landslides* **2012**, *9*, 93–106. [[CrossRef](#)]
55. Cook, A.; Merwade, V. Effect of topographic data, geometric configuration and modeling approach on flood inundation mapping. *J. Hydrol.* **2009**, *377*, 131–142. [[CrossRef](#)]
56. Moore, I.D.; Grayson, R.B. Terrain-based catchment partitioning and runoff prediction using vector elevation data. *Water Resour. Res.* **1991**, *27*, 1177–1191. [[CrossRef](#)]
57. Kirkby, M.; Beven, K. A physically based, variable contributing area model of basin hydrology. *Hydrol. Sci. J.* **1979**, *24*, 43–69.

58. Beven, K.; Kirkby, M.; Schofield, N.; Tagg, A. Testing a physically-based flood forecasting model (topmodel) for three UK catchments. *J. Hydrol.* **1984**, *69*, 119–143. [\[CrossRef\]](#)
59. Glenn, E.P.; Morino, K.; Nagler, P.L.; Murray, R.S.; Pearlstein, S.; Hultine, K.R. Roles of saltcedar (*Tamarix* spp.) and capillary rise in salinizing a non-flooding terrace on a flow-regulated desert river. *J. Arid Environ.* **2012**, *79*, 56–65. [\[CrossRef\]](#)
60. Chung, C.-J.F.; Fabbri, A.G. Validation of spatial prediction models for landslide hazard mapping. *Nat. Hazards* **2003**, *30*, 451–472. [\[CrossRef\]](#)
61. Keršulienė, V.; Zavadskas, E.K.; Turskis, Z. Selection of rational dispute resolution method by applying new step-wise weight assessment ratio analysis (SWARA). *J. Bus. Econ. Manag.* **2010**, *11*, 243–258. [\[CrossRef\]](#)
62. Keršulienė, V.; Turskis, Z. Integrated fuzzy multiple criteria decision making model for architect selection. *Technol. Econ. Dev. Econ.* **2011**, *17*, 645–666. [\[CrossRef\]](#)
63. Zolfani, S.H.; Aghdaie, M.H.; Derakhti, A.; Zavadskas, E.K.; Varzandeh, M.H.M. Decision making on business issues with foresight perspective; an application of new hybrid MCDM model in shopping mall locating. *Expert Syst. Appl.* **2013**, *40*, 7111–7121. [\[CrossRef\]](#)
64. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132. [\[CrossRef\]](#)
65. Jang, J.-S.R. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [\[CrossRef\]](#)
66. Phootrakornchai, W.; Jiriwibhakorn, S. Online critical clearing time estimation using an adaptive neuro-fuzzy inference system (ANFIS). *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 170–181. [\[CrossRef\]](#)
67. Rezakazemi, M.; Dashti, A.; Asghari, M.; Shirazian, S. H<sub>2</sub>-selective mixed matrix membranes modeling using ANFIS, PSO-ANFIS, GA-ANFIS. *Int. J. Hydrogen Energy* **2017**, *42*, 15211–15225. [\[CrossRef\]](#)
68. Jang, J.-S.; Sun, C.-T. Neuro-fuzzy modeling and control. *Proc. IEEE* **1995**, *83*, 378–406. [\[CrossRef\]](#)
69. Chen, W.; Panahi, M.; Pourghasemi, H.R. Performance evaluation of GIS-based new ensemble data mining techniques of adaptive neuro-fuzzy inference system (ANFIS) with genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) for landslide spatial modelling. *Catena* **2017**, *157*, 310–324. [\[CrossRef\]](#)
70. Ahmadlou, M.; Karimi, M.; Alizadeh, S.; Shirzadi, A.; Parvinnejad, D.; Shahabi, H.; Panahi, M. Flood susceptibility assessment using integration of adaptive network-based fuzzy inference system (ANFIS) and biogeography-based optimization (BBO) and BAT algorithms (BA). *Geocarto Int.* **2018**, *1*–21. [\[CrossRef\]](#)
71. Reynolds, R.G. An introduction to cultural algorithms. In Proceedings of the Third Annual Conference on Evolutionary Programming, Singapore, 24 February 1994.
72. Reynolds, R.G. *Cultural Algorithms: Theory and Applications*; McGraw-Hill Ltd.: Maidenhead, UK, 1999; pp. 367–378.
73. Reynolds, R.G.; Ali, M.; Jayyousi, T. Mining the social fabric of archaic urban centers with cultural algorithms. *Computer* **2008**, *41*, 64–72. [\[CrossRef\]](#)
74. Soza, C.; Becerra, R.L.; Riff, M.C.; Coello, C.A.C. Solving timetabling problems using a cultural algorithm. *Appl. Soft Comput.* **2011**, *11*, 337–344. [\[CrossRef\]](#)
75. Pham, D.; Ghanbarzadeh, A.; Koc, E.; Otri, S.; Rahim, S.; Zaidi, M. *The Bees Algorithm. Technical Note*; Manufacturing Engineering Centre, Cardiff University: Wales, UK, 2005; pp. 1–57.
76. Pham, D.; Ghanbarzadeh, A.; Koc, E.; Otri, S.; Rahim, S.; Zaidi, M. The bees algorithm—A novel tool for complex optimisation. *Intell. Prod. Mach. Syst.* **2006**, *454*–459. [\[CrossRef\]](#)
77. Mehrabian, A.R.; Lucas, C. A novel numerical optimization algorithm inspired from weed colonization. *Ecol. Inf.* **2006**, *1*, 355–366. [\[CrossRef\]](#)
78. Ghasemi, M.; Ghavidel, S.; Akbari, E.; Vahed, A.A. Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos. *Energy* **2014**, *73*, 340–353. [\[CrossRef\]](#)
79. Naidu, Y.R.; Ojha, A. A hybrid version of invasive weed optimization with quadratic approximation. *Soft Comput.* **2015**, *19*, 3581–3598. [\[CrossRef\]](#)
80. Zhou, Y.; Luo, Q.; Chen, H.; He, A.; Wu, J. A discrete invasive weed optimization algorithm for solving traveling salesman problem. *Neurocomputing* **2015**, *151*, 1227–1236. [\[CrossRef\]](#)
81. Saravanan, B.; Vasudevan, E.R.; Kothari, D.P. A solution to unit commitment problem using invasive weed optimization algorithm. *Front. Energy* **2013**, *7*, 487–494. [\[CrossRef\]](#)

82. Chen, W.; Peng, J.; Hong, H.; Shahabi, H.; Pradhan, B.; Liu, J.; Zhu, A.-X.; Pei, X.; Duan, Z. Landslide susceptibility modelling using GIS-based machine learning techniques for Chongren County, Jiangxi Province, China. *Sci. Total Environ.* **2018**, *626*, 1121–1135. [[CrossRef](#)] [[PubMed](#)]
83. Shahabi, H.; Hashim, M.; Ahmad, B.B. Remote sensing and GIS-based landslide susceptibility mapping using frequency ratio, logistic regression, and fuzzy logic methods at the central Zab basin, Iran. *Environ. Earth Sci.* **2015**, *73*, 8647–8668. [[CrossRef](#)]
84. Shirzadi, A.; Bui, D.T.; Pham, B.T.; Solaimani, K.; Chapi, K.; Kavian, A.; Shahabi, H.; Revhaug, I. Shallow landslide susceptibility assessment using a novel hybrid intelligence approach. *Environ. Earth Sci.* **2017**, *76*, 60. [[CrossRef](#)]
85. Shahabi, H.; Hashim, M. Landslide susceptibility mapping using GIS-based statistical models and remote sensing data in tropical environment. *Sci. Rep.* **2015**, *5*. [[CrossRef](#)] [[PubMed](#)]
86. Lee, S. Comparison of landslide susceptibility maps generated through multiple logistic regression for three test areas in korea. *Earth Surf. Process. Landf. J. Br. Geomorphol. Res. Group* **2007**, *32*, 2133–2148. [[CrossRef](#)]
87. Pradhan, B. A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using GIS. *Comput. Geosci.* **2013**, *51*, 350–365. [[CrossRef](#)]
88. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]
89. Beasley, T.M.; Zumbo, B.D. Comparison of aligned friedman rank and parametric methods for testing interactions in split-plot designs. *Comput. Stat. Data Anal.* **2003**, *42*, 569–593. [[CrossRef](#)]
90. Kantardzic, M. *Data Mining: Concepts, Models, Methods, and Algorithms*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2011.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Building ANN-Based Regional Multi-Step-Ahead Flood Inundation Forecast Models

Li-Chiu Chang <sup>1,\*</sup>, Mohd Zaki M. Amin <sup>2,\*</sup>, Shun-Nien Yang <sup>1</sup> and Fi-John Chang <sup>3,\*</sup>

<sup>1</sup> Department of Water Resources and Environmental Engineering, Tamkang University, New Taipei City 25137, Taiwan; aa22814946@yahoo.com.tw

<sup>2</sup> Water Resources and Climatic Change Research Centre, National Hydraulic Research Institute of Malaysia, Selangor 43300, Malaysia

<sup>3</sup> Department of Bioenvironmental Systems Engineering, National Taiwan University, Taipei 10617, Taiwan

\* Correspondence: changlc@mail.tku.edu.tw (L.-C.C.); zaki@nahrim.gov.my (M.Z.M.A.); changfj@ntu.edu.tw (F.-J.C.); Tel.: +886-2-3366-3452 (F.-J.C.)

Received: 4 July 2018; Accepted: 13 September 2018; Published: 19 September 2018

**Abstract:** A regional inundation early warning system is crucial to alleviating flood risks and reducing loss of life and property. This study aims to provide real-time multi-step-ahead forecasting of flood inundation maps during storm events for flood early warnings in inundation-prone regions. For decades, the Kemaman River Basin, located on the east coast of the West Malaysia Peninsular, has suffered from monsoon floods that have caused serious damage. The downstream region with an area of approximately 100 km<sup>2</sup> located on the east side of this basin is selected as the study area. We explore and implement a hybrid ANN-based regional flood inundation forecast system in the study area. The system combines two popular artificial neural networks—the self-organizing map (SOM) and the recurrent nonlinear autoregressive with exogenous inputs (RNARX)—to sequentially produce regional flood inundation maps during storm events. The results show that: (1) the 4 × 4 SOM network can effectively cluster regional inundation depths; (2) RNARX networks can accurately forecast the long-term (3–12 h) regional average inundation depths; and (3) the hybrid models can produce adequate real-time regional flood inundation maps. The proposed ANN-based model was shown to very quickly carry out multi-step-ahead forecasting of area-wide inundation depths with sufficient lead time (up to 12 h) and can visualize the forecasted results on Google Earth using user devices to help decision makers and residents take precautionary measures against flooding.

**Keywords:** ANN-based models; flood inundation map; self-organizing map (SOM); recurrent nonlinear autoregressive with exogenous inputs (RNARX)

## 1. Introduction

Floods are the most common natural disasters, and the increasing trend of flood occurrence has been frequently reported worldwide over the last few decades [1–3]. As climate change continues and extensive urbanization worsens, the situation will persist in the vicinities of rivers [4]. Asia and the Pacific Region are the most disaster-prone areas in the world, where floods are the most frequent disasters and have large economic impacts on the region [5]. Recent disasters in Southeast Asia—Typhoon Haiyan (Yolanda) in Philippines (2013) and massive floods in Thailand (2011)—have necessitated the demand for effective flood management schemes, as annual damage estimates far outstretch current management expenditure. The societal cost-benefit analysis of a flood forecasting scheme, compared to a flood management scheme, is different from site to site. Early flood warnings with sufficient lead time offer authorities as well as residents precautions and preventive measures to alleviate consequences and minimize negative impacts. Consequently, there is the increasing need for flood forecasting with lead-times long enough to allow for flood mitigation actions.

Additionally, a real-time flood inundation map showing inundated areas and the corresponding inundating depths could convey a clear message about the flood's scope and severity, thus enabling decision makers and residents to take adequate countermeasures. Subject to various influential factors like rainfall, soil moisture, river-stage conditions, and geomorphological characteristics, as well as a lack of sequential flood inundation datasets during storm events, real-time urban flood inundation forecasting is a great challenge [6].

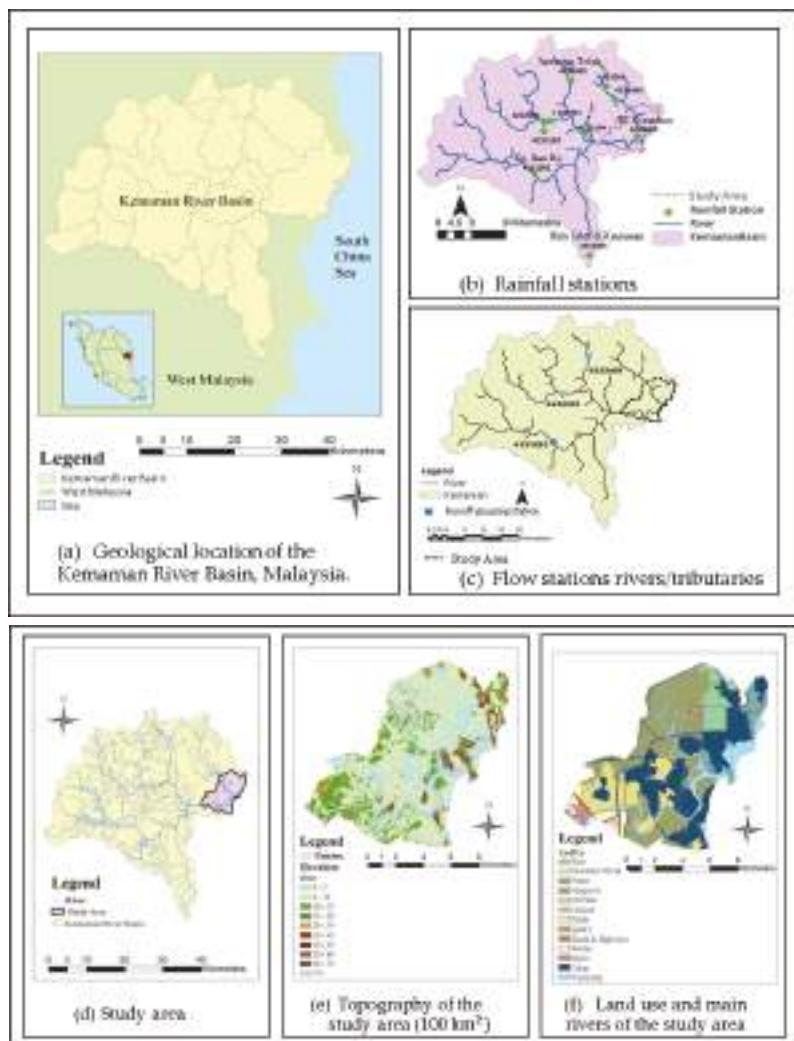
Several studies have been devoted to developing flood inundation simulation models suitable for urban flood management and providing sequential high-resolution inundation depths responsive to storm events [7–9]. Nevertheless, these models required substantial computational efforts in order to deliver iterative solutions and thus could not effectively make real-time inundation forecasts. Practically, hydrologists need to deal with the limited response time to flash flood disasters in urban areas, and therefore multi-step-ahead flood inundation forecasting could be very helpful in managing contingencies and alleviating flood risks. Researches have now moved on from mathematical simulation modelling or physical-based flood forecasting methods to collaboration between methodologies focusing on data-mining approaches and/or physical-based modellers [10,11]. In the last two decades, many studies have developed and/or implemented machine-learning techniques, e.g., artificial neural networks (ANNs) for typhoon flood forecasting [12], runoff forecasting [13,14], real-time multi-step-ahead water level forecasting [15], classification of regional groundwater variations and inter-relations among variables [16], and building multi-relations between fish species and water quality [17]; support vector machines for river flow prediction [18]; fuzzy logic for flow forecasting [19], real-time flood forecasting [20], and water level forecasting [21]; extreme learning machine for stream-flow forecasting [22]; and non-dominated sorting genetic algorithm-II for optimizing water utilization and hydropower output without minimizing flood risks [23]. ANNs generalize relationships between input and output patterns without requiring the physical mechanism behind the process, which is suitable for effectively handling regional rainfall-inundation datasets [24–27]. However, we notice that most of the previous works focused on one-dimensional processes and only very limited studies attempted two-dimensional flood inundation forecasts due to highly dynamic complex spatio-temporal processes involved. Thus, there is a continuous need to conduct in-depth research in regional flood inundation disaster management using the latest scientific tools such as machine learning techniques.

This study aims at establishing a flood early warning system for inundation-prone regions of the Kemaman River Basin in Malaysia. The study basin and its vicinities have frequently suffered from monsoon floods, impacting the socio-economic, agricultural and sub-urban areas of the settlement. We propose an integrated methodology that couples flood inundation simulation and machine learning models to create real-time multi-step-ahead flood inundation maps during storm events. The forecasted regional flood inundation map can be shown on Google Earth to help decision-makers and residents take necessary countermeasures against flooding.

## 2. Study Area and Materials

Floods in Malaysia have great social-economic impacts, and are considered threatening. Being located in the southern part of Terengganu on the east coast of West Malaysia (Figure 1a), the Kemaman River Basin is a region that experiences frequent and severe floods. Flooding has been a major issue in the study catchment since the early 1990s. Storm events result in flooding with short concentration times and a rise in the power and speed of the river flow, causing severe damage to settlements and infrastructures, especially in downstream regions. For instance, the concentration time for Event 7 was less than 12 h and the maximum flow rate reached  $1142 \text{ m}^3/\text{s}$  (Table 1). Therefore, the downstream region that covers an area of approximately  $100 \text{ km}^2$  located on the east side of the Kemaman River Basin is selected as the study area for building real-time regional multi-step-ahead flood inundation forecast models (Figure 1e,f).

The climate of the study basin is generally tropical monsoonal—uniform temperature around 26 °C, high humidity (70–95%) and average annual rainfall of 2770 mm. Historical rainfall and water level data of 10 rainfall and three runoff gauge stations spreading over the study area were collected from the Department of Irrigation and Drainage (DID) of Malaysia; they were used for model calibration and verification. Based on the availability of rainfall and discharge data, eight recent extreme flood events occurring between 2000 and 2015 were selected for this study, as shown in Table 1. The rainfall and river flow stations in the Kemaman River Basin are illustrated in Figure 1b,c, respectively.



**Figure 1.** Information of Kemaman River Basin in Malaysia: (a) geological location, (b) rainfall stations and (c) river flow stations (St.), (d) study area, (e) topography of the study area, and (f) land use and main rivers.

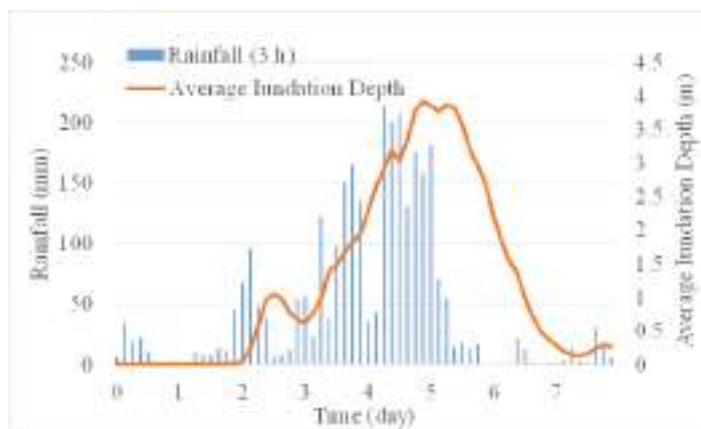
**Table 1.** Real storm events and their characterization.

Event	Beginning (yy/mm/dd)	Ending (yy/mm/dd)	Duration (h)	Maximum Flow (m <sup>3</sup> /s)			Accumulated Average Rainfall (mm)	Maximum Average Inundation Depth (m)
				St*. #4131453	St. #4232401	St. #4332401		
1	2001/12/20, 5:00 a.m.	2001/12/27, 9:00 a.m.	172	731	814	84	453	1.70
2	2003/12/5, 12:00 p.m.	2003/12/12, 9:00 a.m.	165	554	433	183	396	1.67
3	2006/12/20, 2:15 a.m.	2006/12/26, 8:30 a.m.	150	534	207	87	268	1.48
4	2008/12/31, 19:30 a.m.	2009/1/8, 7:30 a.m.	180	478	487	96	390	1.74
5	2012/1/10, 7:15 a.m.	2012/1/17, 4:00 p.m.	177	453	506	39	255	1.70
6	2012/12/13, 0:15 a.m.	2013/1/13, 0:00 a.m.	744	509	920	75	973	3.28
7	2013/11/28, 9:30 p.m.	2013/12/6, 9:30 p.m.	192	745	1142	63	996	4.17
8	2014/12/13, 12:00 p.m.	2015/1/1, 4:00 p.m.	460	634	801	57	1653	3.35

\* River flow station.

The ranges of event duration, maximum average inundation depth, and accumulated average rainfall were recorded as 150 h, 744 h, 1.48 m, 4.17 m; and 255 mm, 1653 mm, respectively. The highest accumulated flood depth (4.17 m) occurred in Event 7, where the accumulated average rainfall reached 996 mm and the maximum flow achieved 1142 m<sup>3</sup>/s. In this study, rainfall in a flood depth hydrograph denotes the average rainfall of each sub-catchment at a time-step of 3 h.

Taking Event 7 as an example, Figure 2 illustrates the rainfall histogram and the average inundation depth hydrograph of Event 7.

**Figure 2.** Rainfall histogram and average inundation depth hydrograph for Event 7.

### 3. Forecast Models

To establish a flood early warning system, we propose an integrated methodology that couples flood inundation simulation models and machine learning-based forecast models to provide multi-step-ahead flood inundation maps during storm events. The hybrid modelling process explored in this study consists of five stages: (1) dividing the study area (100 km<sup>2</sup>) into 10,744 grids (75 m × 75 m); (2) synthesizing the sequential regional inundation depths for real and designed events by using a two-dimensional flood routing simulation model; (3) clustering regional inundation maps by using the self-organizing map (SOM) algorithm; (4) building a forecast model by using the recurrent nonlinear autoregressive with exogenous inputs (RNARX) for obtaining the average regional inundated depth

(ARID) under rainy conditions; and (5) adjusting the weight of the selected neuron in the SOM based on the forecasted ARID to obtain a real-time adapted regional inundation map. It is noted that we removed the cells covering river channels and the cells with highest observed inundation depths less than 10 cm (for example, inundation did not occur at cells at relatively high elevations). Consequently, about 40% of the cells were removed, i.e., the number of grids reduced from 17,778 to 10,740. Thus, the input dimension (number of grids) of the SOM largely reduced, such that the model (SOM) is easier to train and converge. Besides, because the study area is located downstream (estuary) and its terrain in general is relatively flat, we consider the designed grid size (i.e., 75 m × 75 m) suitable for presenting flood inundation variation during storm periods.

The average regional inundation depth (ARID) is calculated by the following equation:

$$ARID = \frac{\sum \text{inundation depth in each grid}}{\text{the number of grids}} \quad (1)$$

These models are briefly introduced in the following subsections.

### 3.1. Simulation Model

Historical observed data of inundation depths are rarely available, such that there are almost no historical hydrographs of regional inundation depths of historical storm events. Hydraulic models, in general, are a more physical-based approach to modeling the motion of a water body through its geo-morphological environment, taking into account the effects of gravity and friction at the water/bed interface to produce flood levels at various locations along the river and the flood plain [28]. Therefore, the flood routing simulation model would be the best alternative for generating the synthetic hydrographs of flood depths for various storm events; it will help understand the induced flood hazards and identify the responses of mitigation measures. InfoWorks ICM (ICM Suite upgrade, HR Wallingford Asia Sdn Bhd, Kuala Lumpur, Malaysia), a commercial software package and an advance of Infoworks RS, is a two-dimensional flood routing simulation tool that combines an advanced flow simulation engine, hydrological and hydraulic models, GIS, and database in a single environment [28]. This model relies profoundly on the accuracy of topographical data [29]. InfoWorks ICM (RS) has been satisfactorily utilized to develop flood risk maps for Northern Ireland [30] and Malaysia's River Basin [31]. This tool offers explanations for the effects of varying runoff rates from sub-catchments and forms a tight-coupling of GIS functionality and hydrodynamic flow simulation.

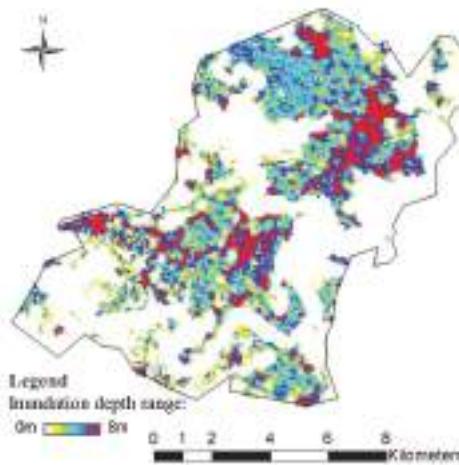
In this study, InfoWorks ICM is implemented to mimic the effects of flood mitigation measures under various real and synthetic events. The main components of the InfoWorks ICM consist of an ISIS flow simulation engine, a geographic information system (GIS), and a database. The model carries out two-dimensional flood routing simulation in consideration of hydraulic facilities/modeling [32,33]. The two-dimensional (2D) hydrodynamic model routes water over in the Digital Elevation Model (DEM) forced by upstream boundary conditions such as a meteorological forcing, or a hydrograph and downstream boundary conditions of water levels. The model generates accurate and reproducible flood extent maps, which not only show the extent and depth of a flood, but can also be animated to show the progression of a flood event. The scenario setting for simulating events was designed and described as follows.

- (1) Observed distribution: 8 real events (rainfall data).
- (2) Synthetic distribution: a total of 12 designed events were generated, based on different combinations of rainfall durations (12, 24, 48 and 72 h) and return periods (20, 50 and 100 years) at main gauge stations.

Figure 3 shows the maximum simulated flood map derived from the 2D flood model (InfoWorks ICM) based on a 5 m resolution digital elevation and rainfall information of a storm event (a cumulated rainfall of 1653 mm in December 2014). We note that this simulation model may not necessarily yield the exact flood depths but can provide the magnitudes of water depths and flood

extents. Consequently, the simulation model (InfoWorks ICM) with designed rainfall hyetographs are implemented to generate the regional inundation maps and provide the inundation depth of each grid element for all scenarios. The generated inundation maps could then be used to train and validate ANN-based models.

Coupling the observed and simulated datasets, the RNARX firstly makes 3 h-up to 12 h-ahead forecasts (ARID); next, the forecasted ARID can be mapped onto the SOM topological map to select the best matched cluster in the SOM, and then all the weights in the matched cluster can be adjusted using linear interpolation to obtain a 3 h- (or, 6 h-, 9 h-, 12 h-) ahead regional flood inundation map.



**Figure 3.** Maximum simulated flood map derived from the 2D flood model (a rainfall of 1653 mm in December 2014).

### 3.2. ANN-Based Models

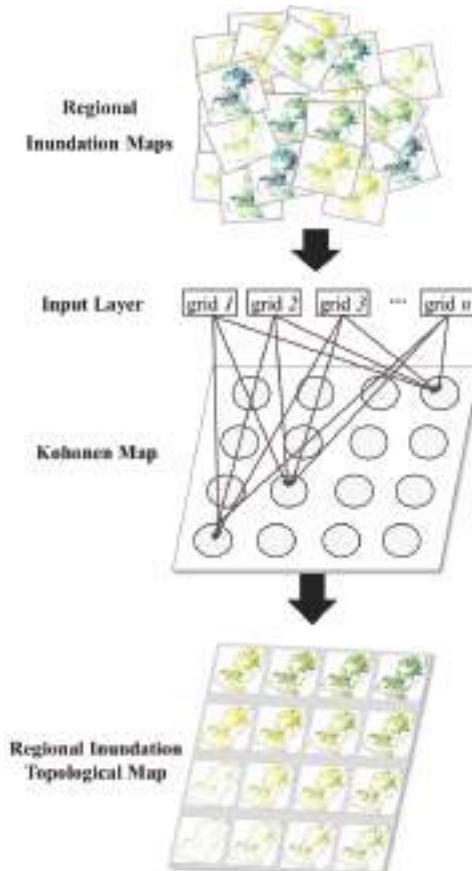
In this study, we explore a hybrid SOM–RNARX model that integrates the SOM and the RNARX to sequentially forecast regional multi-step-ahead flood inundation maps. The key idea is to integrate the essence of both ANNs to produce two-dimensional visible regional flood inundation maps.

#### 3.2.1. Self-Organizing Map (SOM)

The SOM is an effective clustering method that can classify high-dimensional datasets to form a meaningful topological map; it offers the advantages of information extraction and visualization [34]. For instance, the SOM was implemented as a clustering tool to classify inundation maps [35], groundwater levels [36,37], fishery data [17,38], and river flow stations [39]. The SOM consists of an input layer and a clustering layer with nodes distributed over a two-dimensional map. Bearing in mind its clustering capability and visual interpretation, the SOM in this study is implemented to form a two-dimensional topological map that presents the main features of regional flood inundation depths and extends under various hydrological conditions during storm events (Figure 4). The map size of the SOM must be determined, which is usually done by trial-and-error procedures, because there are no theoretical guidelines to determine optimal map size [40]. Due to the large input dimension (i.e., 10,744 grids) and relatively stable variations of regional inundation depths, we only try three different map sizes (i.e.,  $3 \times 3$ ,  $4 \times 4$ , and  $5 \times 5$ ) to form regional inundation topological maps in this study case.

### 3.2.2. Recurrent Configuration of Nonlinear Autoregressive with Exogenous Inputs (RNARX)

The nonlinear autoregressive with exogenous input (NARX) network is a powerful dynamic model for time series prediction [41]. Several studies have demonstrated that the recurrent configuration of the RNARX network (RNARX) could suitably forecast time series with long-range dependence present in data [42–45]. Besides, there is a common lack of real-time observed inundation depths, which is a great challenge for building a reliable real-time model. To solve the problem, the RNARX network offers the advantage of using model outputs as parts of inputs to train the model and use the configured model for real-time forecasting in test (or real) cases. To tackle the problem of no actual real-time observations available during model construction, the recurrent scheme of the RNARX network utilizes model outputs as parts of model inputs in training, validation and testing stages. That is to say, this recurrent network could be trained with imperfect information but similar characteristics of input-output patterns in all three phases (i.e., training, validation and testing phases), such that the configured network would maintain a similar forecasting capability in all phases. The constructed network and its synaptic weights would be fixed in the validation and testing phases to evaluate its reliability with new events. For multi-step-ahead forecasting tasks, current forecast values could be sequentially fed back to the input layer to provide one-step further forecasts.



**Figure 4.** Cluster the spatial distributions of inundation maps by the SOM ( $4 \times 4$  neurons).

Assuming the unit time delay and the output-memory order are given, this nonlinear function of the RNARX can be expressed by Equation (2):

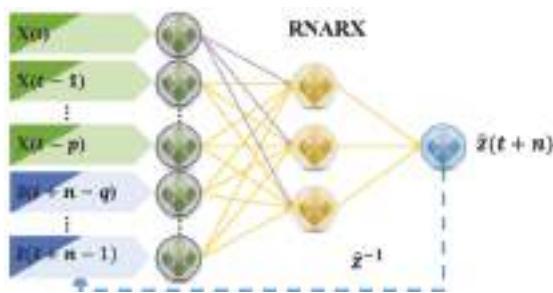
$$z(t) = f[z(t-1), \dots, z(t-q); U(t)] \quad (2)$$

where  $U(t)$  and  $z(t)$  are the input vector and output value at a time step  $t$ , respectively, and  $f[\cdot]$  is the nonlinear function,  $q$  denotes the number of time steps.

The RNARX is commonly trained in two modes: parallel (P) and series-parallel (SP) modes. In the P mode, the forecasted outputs ( $z(t-i)$ ,  $i = 1 - q$ ) are fed back into the input layer, represented by Equation (2). In the SP mode, the output regressor in the input layer is formed by observational values ( $d(t-i)$ ,  $i = 1 - q$ ), as shown in Equation (3):

$$z(t) = f[d(t-1), \dots, d(t-q); U(t)] \quad (3)$$

The architecture of the RNARX model is illustrated in Figure 5.



**Figure 5.** Architecture of the RNARX network.  $X$  and  $\hat{Z}$  denote observed and forecasted values, respectively.

In this study, we implement the RNARX for flood forecasting. The RNARX model is configured by using model outputs as parts of model inputs, and its weights are adjusted by using the conjugate gradient back-propagation learning algorithm to search for the minimum errors during network training.

### 3.3. ANNs-Based Regional Inundation Depth Forecasting

We next discuss how to integrate the SOM and the RNARX for forecasting regional flood inundation maps. The hybridizing process consist of three major schemes (Figure 6). The first scheme involves clustering regional inundation maps using the SOM. The second scheme involves building a forecast model using the RNARX to obtain the ARID under rainy conditions. The third scheme involves adjusting the weight of the selected neuron in the SOM based on the forecasted ARID to obtain a real-time adapted regional inundation map. To be more specific, the ARID of a neuron in the SOM can be obtained by weightedly summing (averaging) the inundation depths in all grids of the neuron, and then the neuron with a value that is the closest to the forecasted value is selected as the best matched (selected) neuron. The ARID of the selected (winning) neuron can be further modified by a linear interpolation method to adjust all the regional inundation depths in the neuron. Using this approach, the regional inundation maps stored in the SOM can be instantly updated during storm events, and the accuracy of the forecasted inundation map will significantly increase. We would like to note that real-time rainfall information might be the only information available for modelling in real applications, and therefore the proposed methodology does provide an effective and promising way for real-time forecasting regional flood inundation maps and their extents. The proposed methodology can be repeatedly implemented for obtaining multi-step-ahead forecasts.

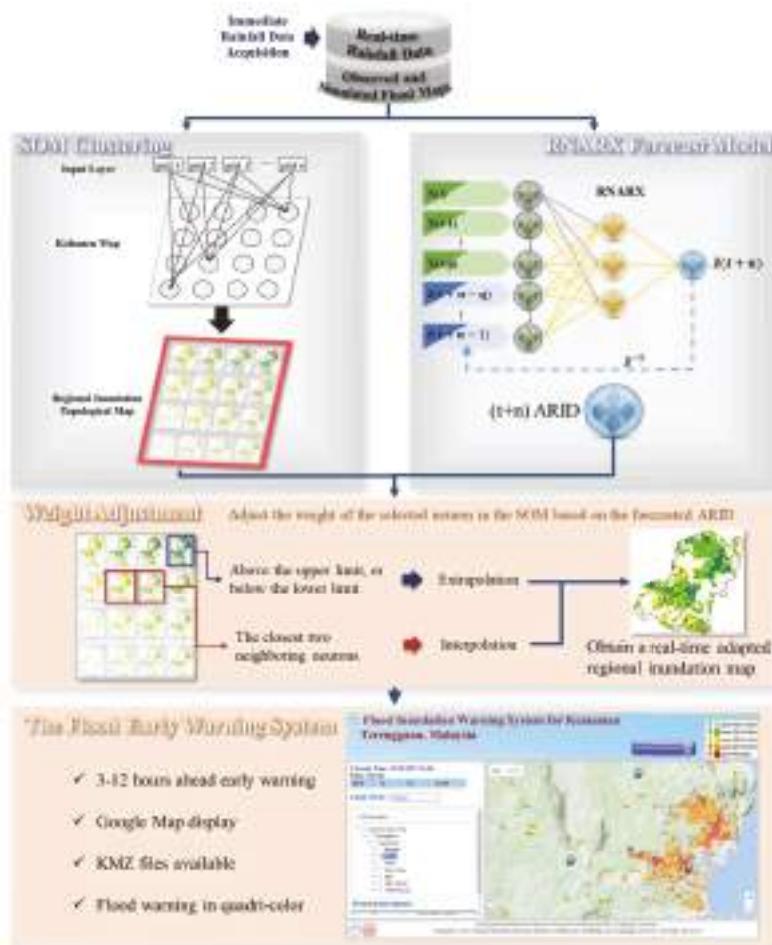
### 3.4. Evaluation Indexes

The performance of the proposed approach is evaluated by the coefficient of determination ( $R^2$ ) and the root mean square error (RMSE), as shown below:

$$R^2 = \left[ \frac{\sum_{i=1}^N (D_i - \bar{D})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^N (D_i - \bar{D})^2 \sum_{i=1}^N (Y_i - \bar{Y})^2}} \right]^2 \quad (4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (D_i - Y_i)^2}{N}} \quad (5)$$

where  $D_i$  and  $Y_i$  are the forecasted value and the observed (simulated) value of the  $i$ th data, respectively.  $\bar{D}$  and  $\bar{Y}$  are the mean values of the forecasted values and the observed values, respectively.



**Figure 6.** Methodology of the hybrid SOM–RNARX model.

## 4. Results and Discussion

### 4.1. SOM Clustering

There are 225 (relevant to Events 1, 3, 5 and 7) datasets (i.e., regional inundation depths) to be classified into the neurons of the SOM in the training phase. Each neuron in the SOM contains 10,774 grids distributed over the study area. The classification results of the spatial distributions of inundation maps are shown in Figure 7. We find that the SOM networks with map sizes of  $4 \times 4$  can produce well the clustering topology of regional inundation depths. The two-dimensional ( $4 \times 4$ ) topological map clearly presents the overall results of the neurons, which exhibits the variation trend of regional inundation depths ranging from [0–1 m] to [0–over 7 m]. The topological relationships between individual neurons and their neighboring neurons are visible and distinguishable. In Figure 7, the ARID of each neuron is shown at the top right corner of the neuron. The bottom-left section of each neuron is light yellow in color, showcasing the lowest input values (ARID = 0.055 m), while the top-right section of each neuron is dark blue in color, denoting the highest input values (ARID = 3.439 m).

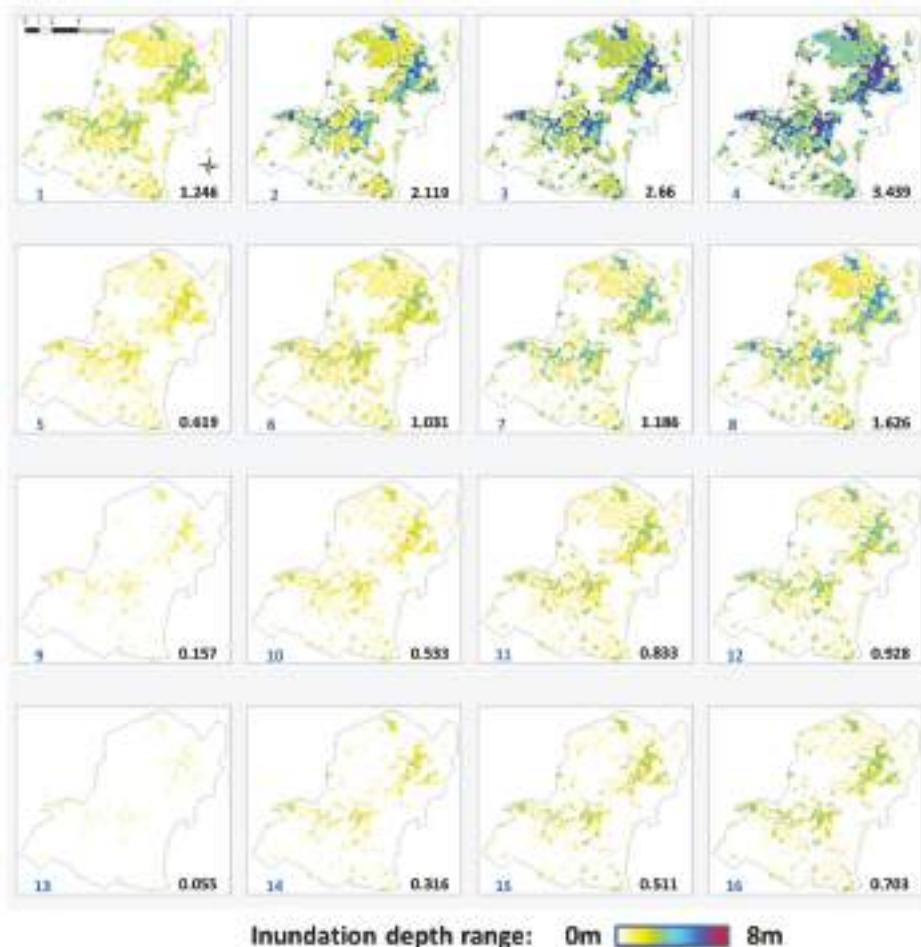


Figure 7. Topology of  $4 \times 4$  clustering maps.

The constructed topological map can easily display the difference between these two corners (bottom-left neuron and top-right neuron); moreover, the topological images of the neurons nearby each of these two corners are quite similar. These figures show that the SOM network can effectively cluster high-dimensional (10,744 grids) data to extract (and present) meaningful topological structures.

#### 4.2. Forecasting Average Regional Inundation Depths (ARID)

Four ARID forecasting models are individually constructed for three- up to twelve-hour-ahead forecasting by using RNARX networks based on a large number of rainfall-total inundated volume patterns. There are 7 real events and 12 designed events used to build and/or test the forecast models. Data allocation for model construction is shown in Table 2. The inputs consist of the current observed three-hour rainfall of three sub-catchments and the recurrently forecasted ARIDs at time step of  $T + n - 1$  while the output is the  $n$ -step-ahead ( $T + n$ ) ARID. Table 3 shows the input combinations of the RNARX models at various time steps. For instance, the input dimension is 4 (three observed values and one recurrent forecasted value) and the output dimension is only 1 for  $T + 4$  (12 h-ahead). After implementing an intensive trial-and-error procedure based on the training and validation datasets, the most suitable RNARX network is identified to have only one hidden layer with five nodes, which would in general produce admirable forecast performances.

**Table 2.** Dataset allocation for constructing RNARX models.

Dataset	Real Events	Designed Events
Training	Event 3	100yearreturnperiod_12hourRainfall
	Event 5	50yearreturnperiod_24hourRainfall
	Event 6	20yearreturnperiod_48hourRainfall
	Event 7	20yearreturnperiod_72hourRainfall
		100yearreturnperiod_72hourRainfall
Validation	Event 2	20yearreturnperiod_12hourRainfall
	Event 4	20yearreturnperiod_24hourRainfall 100yearreturnperiod_48hourRainfall
Testing	Event 8	50yearreturnperiod_12hourRainfall
		100yearreturnperiod_24hourRainfall
		50yearreturnperiod_72hourRainfall

**Table 3.** Input combinations for the RNARX model.

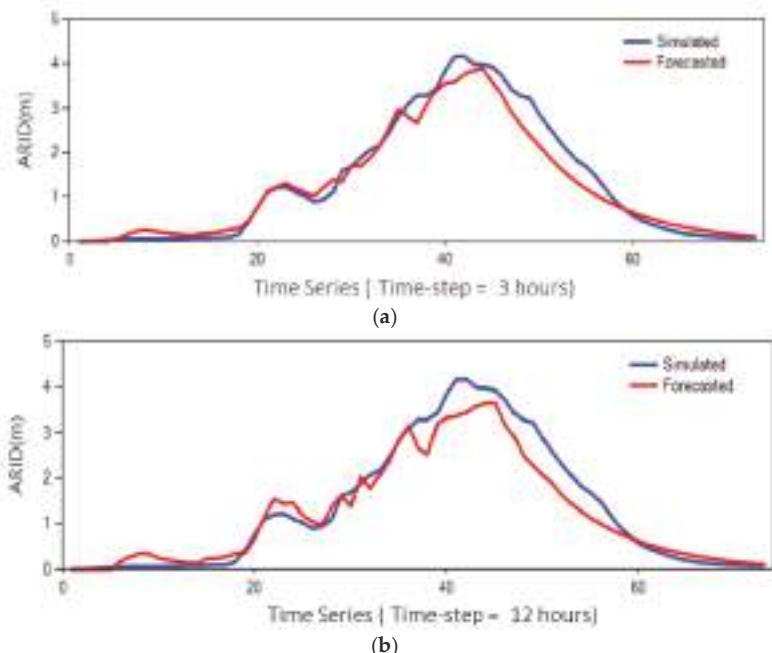
Forecast Time-Step \ Input Factors	Kemaman River Average Rainfall	Cherul River Average Rainfall	Flood Depth
$T + 1$ (3 h-ahead)	$T - 2$	$T - 2$	-
$T + 2$ (6 h-ahead)	$T - 1$	$T - 1$	-
$T + 3$ (9 h-ahead)	$T$	$T$	-
$T + 4$ (12 h-ahead)	$T$	$T$	$T + 3$

Table 4 shows the summarized RNARX results for 3 h-up to 12 h-ahead forecasting of ARID in terms of RMSE and  $R^2$ . The  $R^2$  values of 3–9 h ahead forecasting are greater than 0.90 in these three phases, which indicate that the constructed models can well forecast ARIDs with very high correlation, while those of 12 h-ahead forecasting are also greater than 0.87. It appears that the models can forecast ARIDs accurately and reliably. We notice that the 3 h-ahead, 6 h-ahead and 9 h-ahead forecasting models produce very similar results because these three models have the same inputs and parameters.

**Table 4.** Performance of 3 h-up to 12 h-ahead forecasting of ARID using RNARX based on 7 real events and 12 designed events.

Forecasting Time-Step	RMSE (m)			$R^2$		
	Training	Validation	Testing	Training	Validation	Testing
3 h-ahead	0.28	0.30	0.34	0.92	0.90	0.90
6 h-ahead	0.28	0.30	0.34	0.92	0.90	0.90
9 h-ahead	0.29	0.30	0.33	0.91	0.90	0.90
12 h-ahead	0.31	0.34	0.35	0.90	0.87	0.89

Figure 8 illustrates the simulated and forecasted ARIDs obtained from the RNARX model at two different horizons for Event 7. The results indicate that the RNARX perform well at both shorter forecast horizon (3 h) and longer forecast horizon (12 h). Therefore, it is suitable to use the RNARX to make forecasts on ARIDs.



**Figure 8.** Simulated and forecasted ARIDs obtained from the RNARX model for Event 7. (a) 3 h-ahead simulated and forecasted ARIDs; (b) 12 h-ahead simulated and forecasted ARIDs.

#### 4.3. ANNs-Based Models for Forecasting Regional Inundation Maps

In this hybrid stage, the forecasted ARID of the RNARX network is used to adjust the selected SOM neuron whose ARID is the closest to the forecasted ARID of the RNARX network. Table 5 shows the results of the simulation model for Event 7 in the study area (10,744 grids) with respect to different inundation depths (i.e., All, 0–1 m, 1–2 m, 2–3 m, and >3 m) and time steps ranging from the beginning of flooding at  $T = 75$  to the end of flooding at  $T = 120$ . As shown, when  $T = 75$  and  $T = 81$ , more than half of the grids have small inundation depths (i.e., 0–1 m) and the numbers of grids for higher inundation depths (i.e., 2–3 m and over 3 m peak) gradually increase over time. When  $T = 105$ , almost

half the grids have inundation depths exceeding 3 m, and ARIDs reach the maximum at  $T = 120$  and then decrease toward the end of flooding.

**Table 5.** Numbers of grids with respect to different inundation depths obtained from the simulation model over time for Event 7.

Time (h)	All	0–1 m	1–2 m	2–3 m	>3 m
$T = 75$	10,744	7274	1361	1231	908
$T = 81$	10,744	6530	1617	1376	1251
$T = 87$	10,744	4488	2176	1964	2146
$T = 93$	10,744	3466	2053	2452	2803
$T = 108$	10,744	1400	1195	2473	5706
$T = 120$	10,744	614	780	1707	7673

Table 6 shows the forecast performance ( $R^2$  and RMSE) of the hybrid SOM–RNARX model with respect to different inundation depths over time for Event 7, taking the simulation model as a benchmark. We find that the forecast model in general can provide very accurate and reliable results, in terms of small RMSE values and very high  $R^2$  values.

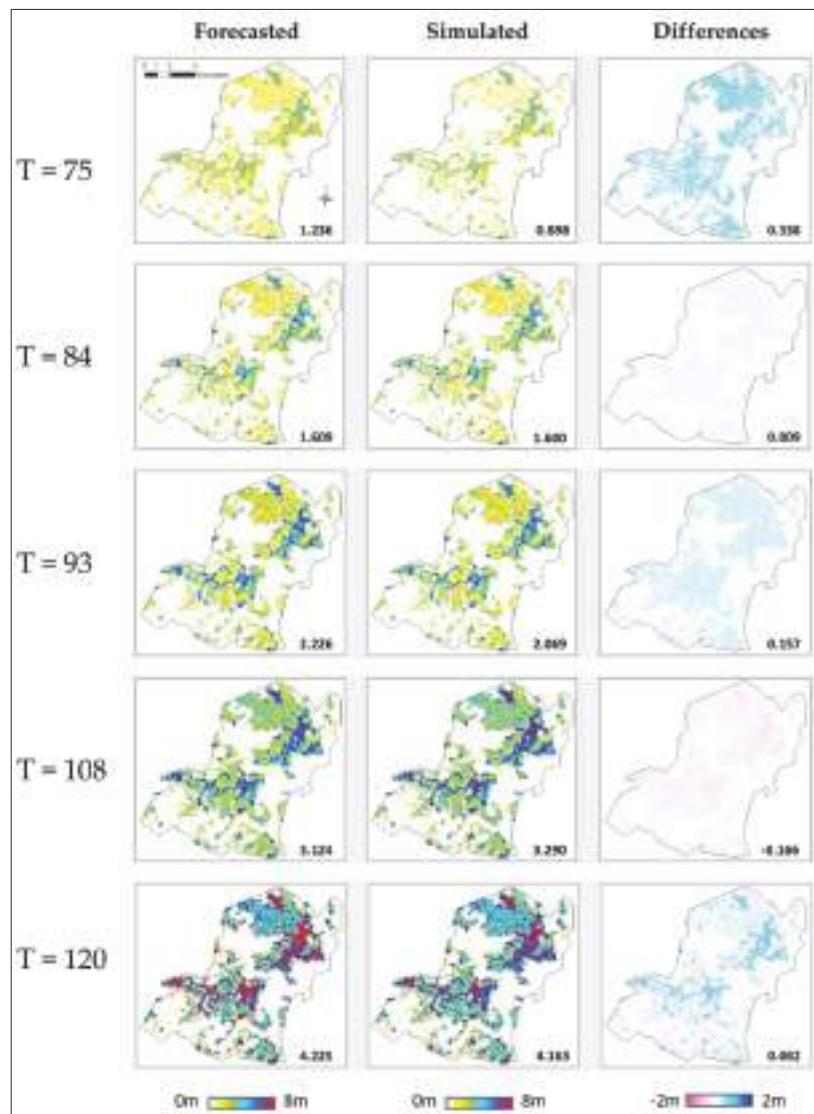
**Table 6.** Performances of the SOM–RNARX model (time step = 3 h) with respect to different inundation depths over time for Event 7, taking the simulation model as a benchmark.

Time (h)	Time Step	All		0–1 m	1–2 m	2–3 m	>3 m
		RMSE *	$R^2$	RMSE	RMSE	RMSE	RMSE
$T = 75$	$T + 1 (=3 \text{ h})$	0.45	0.94	0.52	0.38	0.16	0.09
	$T + 2 (=6 \text{ h})$	0.44	0.94	0.51	0.36	0.14	0.10
	$T + 3 (=9 \text{ h})$	0.45	0.94	0.52	0.38	0.17	0.09
	$T + 4 (=12 \text{ h})$	0.49	0.94	0.55	0.45	0.25	0.17
$T = 81$	$T + 1$	0.65	0.97	0.53	0.79	0.81	0.84
	$T + 2$	0.64	0.97	0.52	0.77	0.79	0.81
	$T + 3$	0.66	0.97	0.53	0.79	0.82	0.85
	$T + 4$	0.68	0.97	0.54	0.81	0.85	0.90
$T = 87$	$T + 1$	0.38	0.99	0.36	0.45	0.40	0.34
	$T + 2$	0.37	0.99	0.35	0.43	0.39	0.31
	$T + 3$	0.39	0.99	0.36	0.45	0.41	0.35
	$T + 4$	0.08	1	0.06	0.08	0.08	0.12
$T = 93$	$T + 1$	0.19	1	0.16	0.2	0.2	0.19
	$T + 2$	0.19	1	0.10	0.16	0.19	0.19
	$T + 3$	0.19	1	0.16	0.21	0.21	0.19
	$T + 4$	0.12	1	0.09	0.12	0.13	0.12
$T = 108$	$T + 1$	0.25	1	0.11	0.07	0.12	0.32
	$T + 2$	0.25	1	0.11	0.07	0.12	0.32
	$T + 3$	0.24	1	0.11	0.07	0.11	0.31
	$T + 4$	0.18	1	0.12	0.08	0.08	0.23
$T = 120$	$T + 1$	0.42	1	0.48	0.57	0.34	0.42
	$T + 2$	0.42	1	0.48	0.57	0.34	0.42
	$T + 3$	0.42	1	0.48	0.57	0.34	0.42
	$T + 4$	0.34	1	0.49	0.61	0.43	0.34

\* Unit: m.

Figure 9 shows the regional flood inundation depths over time (from  $T = 75$  to  $T = 120$ ) for Event 7 in consideration of (1) forecasted results, (2) simulated results, and (3) differences between the forecasted and simulated results. We can find that the inundation depths range from low value ( $T = 75$ ) to high value ( $T = 120$ ) and the forecasted results agree with the simulated results. It appears that the proposed

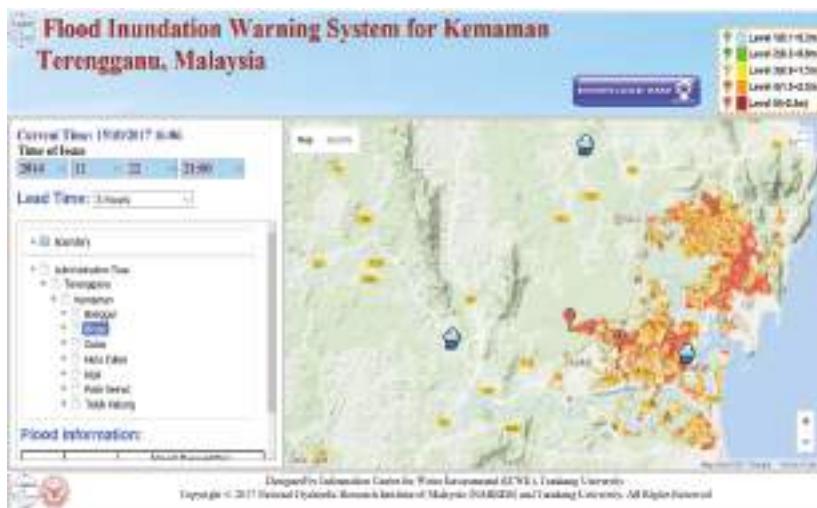
model could perform well because the differences in all the grids of the region are relatively small (less than 0.3 meter in most regions and time-steps). We conclude that the constructed models can produce suitable and reliable multi-step-ahead inundation forecasts and can be used to build a flood early warning system for the study area. We also notice that the proposed methodology can very quickly carry out multi-step-ahead (3 h up to 12 h) forecasting of area-wide inundation depths, with computation time less than few seconds, and thereby leading to real-time regional flood inundation forecasting.



**Figure 9.** Regional flood inundation depths at different time steps ranging from  $T = 75$  (beginning) to  $T = 120$  (ending) for Event 7 in consideration of (1) forecasted results, (2) simulated results, and (3) differences between the forecasted and simulated results, respectively.

#### 4.4. Development of the Flood Early Warning System

The final output of this study is the display of the on-line real-time flood early warning system, where the inundation status in the study area can be visualized online. The system display embeds Google Map in the webpage of this system and provides Keyhole Markup Language (KMZ) files for download. Users can see the regional flood inundation depths and quickly find a location's inundation depth. Figure 10 shows the real-time regional flood inundation forecasting results displayed on the Web. It consists of three parts: Header, Main Page, and Footer. The header composes of the title of the system, the legend and the KMZ download button, which visualize the forecasted results on Google Earth in user devices. The left portion of the main page includes time of issue, lead time, tree views of boundary and administration for selecting the forecasted results of interest on Google Map. The rest of the main page is partitioned into two sub-parts: control panel and Google Map. The legend of the flood inundation level is shown in the upper left corner (Figure 10). A quadri-color flood warning system is built in this study, where blue denotes a warning of Level 1 indicating 0.1–0.3 m flood depths; green denotes a warning of Level 2 indicating 0.3–0.9 m flood depths; yellow denotes a warning of Level 3 indicating 0.9–1.5 m flood depths; orange denotes a warning of Level 4 indicating 1.5–2.5 m flood depths; and red denotes a warning of Level 5 indicating flood depths exceeding 2.5 m. The main function of the display interface is to show the inundation area over Google Map, to display the maximum inundation depth with its location in each sub-district, and to provide a KMZ file download of results that can be browsed on Google Earth.



**Figure 10.** Regional flood inundation information on the designed Web.

#### 5. Conclusions

This study aims at exploring an inundation forecasting approach for future forecast horizons based on the combination of a clustering technique (SOM) of a set of high-resolution inundation maps and an inundation depth forecasting model based on an artificial neural network (RNARX). We then establish a web-based (online) flood early warning system that enables the issuing of an advanced warning of possible flash floods and/or regional inundation depth. We propose a novel hybrid SOM–RNARX model for real-time forecasting of the spatial distribution of flood inundation in the study area of the Kemaman River Basin in Malaysia. We demonstrate that the regional inundation maps stored in the SOM can be instantly updated during storm invasion periods and the accuracy of the forecasted

inundation maps can be significantly increased through real-time ARID forecasts obtained from the RNARX network. The proposed methodology and the primary results are summarized as follows:

- (1) The input datasets for the SOM consist of high-dimensional spatial inundation depths (with a grid resolution of 75 m × 75 m) of the study area obtained from the 2-D simulation model based on a number of real storm events. The main features of the spatial inundation distributions can be well distinguished by an SOM with 4 × 4 neurons to obtain a distinguishable topology. The SOM network can effectively cluster the high-dimensional (10,744 grids) inundation depths to extract and present their topological structures.
- (2) The results suggest that the RNARX network configured with current regional rainfall information and the model's recurrent output can well capture the main features of the input-output patterns to provide stable and reliable forecasts of ARIDs.
- (3) The proposed model integrates the favorable essence of both networks (SOM & RNARX) and fuses their corresponding results to provide real-time visible regional multi-step-ahead flood inundation maps with high resolution; their nowcasts are reliable and adequate (with small RMSE and high R<sup>2</sup> values).
- (4) Regarding the execution efficiency of the developed system for the study area, the system can very quickly (in just a few seconds) carry out three to twelve-hour-ahead forecasting of area-wide inundation maps and thereby lead to real-time flood forecasting.

**Author Contributions:** Conceptualization, F.-J.C., L.-C.C. and M.Z.M.A.; Methodology, L.-C.C., F.-J.C. and M.Z.M.A.; Software, L.-C.C., M.Z.M.A. and S.-N.Y.; Validation, L.-C.C. and M.Z.M.A.; Formal Analysis, F.-J.C., L.-C.C., M.Z.M.A. and S.-N.Y.; Investigation, L.-C.C. and M.Z.M.A.; Resources, F.-J.C., L.-C.C. and M.Z.M.A.; Data Curation, L.-C.C., M.Z.M.A. and S.-N.Y.; Writing—Original Draft Preparation, F.-J.C. and L.-C.C.; Writing—Review & Editing, F.-J.C., L.-C.C. and M.Z.M.A.; Visualization, L.-C.C.; Supervision, F.-J.C. and M.Z.M.A.; Project Administration, L.-C.C. and M.Z.M.A.; Funding Acquisition, M.Z.M.A. and L.-C.C.

**Funding:** This research was funded by Ministry of Water, Land and Natural Resources Malaysia (formerly known as Ministry of Natural Resources and Environment Malaysia through National Hydraulic Research Institute of Malaysia (NAHRIM) in collaboration with Tamkang University of Taiwan.

**Acknowledgments:** The authors are grateful to the National Hydraulic Research Institute of Malaysia (NAHRIM) for supporting and facilitating this study and providing hydrological data and analysis, and performed 2D hydrodynamics modeling as well as produced 2D riverine simulated flood inundation datasets. The authors would like to thank the Editors and anonymous Reviewers for their valuable and constructive comments related to this manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Doocy, S.; Daniels, A.; Murray, S.; Kirsch, T.D. The Human Impact: A Historical Review of Events and Systematic Literature Review. *PLoS Curr. Disasters* **2013**, *1*, 1–32.
2. Guha-Sapir, D.; Hoyois, P.; Below, R. *Annual Disaster Statistical Review 2014: The Numbers and Trends*; CRED, Université Catholique de Louvain: Brussels, Belgium, 2015.
3. Guha-Sapir, D.; Hoyois, P.; Below, R.; Vanderveken, A. *Annual Disaster Statistical Review 2015: The Numbers and Trends*; CRED, Université Catholique de Louvain: Brussels, Belgium, 2016.
4. Balica, S.F.; Popescu, I.; Beavers, L.; Wright, N.G. Parametric and physically based modelling techniques for flood risk and vulnerability assessment: A comparison. *Environ. Model. Softw.* **2013**, *41*, 84–92. [[CrossRef](#)]
5. ESCAP IDD, United Nations. Disasters in Asia and the Pacific: 2015 Year in Review. Available online: [https://www.unescap.org/sites/default/files/2015\\_Year%20in%20Review\\_final\\_PDF\\_1.pdf](https://www.unescap.org/sites/default/files/2015_Year%20in%20Review_final_PDF_1.pdf) (accessed on 1 December 2017).
6. Gourley, J.; Clark, R., III. Real-time flash flood forecasting. In *Oxford Encyclopedia of Natural Hazard Science*; Oxford University Press: Oxford, UK, 2018.
7. Bates, P.D.; Horritt, M.S.; Fewtrell, T.J. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *J. Hydrol.* **2010**, *387*, 33–45. [[CrossRef](#)]
8. Han, S.; Coulibaly, P. Bayesian flood forecasting methods: A review. *J. Hydrol.* **2017**, *551*, 340–351. [[CrossRef](#)]

9. Neal, J.C.; Odoni, N.A.; Trigg, M.A.; Freer, J.E.; Garcia-Pintado, J.; Mason, D.C.; Wood, M.; Bates, P.D. Efficient incorporation of channel cross-section geometry uncertainty into regional and global scale flood inundation models. *J. Hydrol.* **2015**, *529*, 169–183. [[CrossRef](#)]
10. Abrahart, R.J.; See, L.M.; Solomatine, D.P. *Practical Hydroinformatics: Computational Intelligence and Technological Developments in Water Applications*; Springer: Berlin/Heidelberg, Germany, 2008.
11. Mount, N.J.; Maier, H.R.; Toth, E.; Elshorbagy, A.; Solomatine, D.; Chang, F.J.; Abrahart, R.J. Data-driven modelling approaches for social-hydrology: Opportunities and challenges within the Panta Rhei Science Plan. *Hydrol. Sci. J.* **2016**, *61*, 1192–1208.
12. Jhong, Y.D.; Chen, C.S.; Lin, H.P.; Chen, S.T. Physical Hybrid Neural Network Model to Forecast Typhoon Floods. *Water* **2018**, *10*, 632. [[CrossRef](#)]
13. Badrzadeh, H.; Sarukkalige, R.; Jayawardena, A.W. Hourly runoff forecasting for flood risk management: Application of various computational intelligence models. *J. Hydrol.* **2015**, *529*, 1633–1643. [[CrossRef](#)]
14. Yu, Y.; Zhang, H.; Singh, V.P. Forward prediction of runoff data in data-scarce basins with an improved ensemble empirical mode decomposition (EEMD) model. *Water* **2018**, *10*, 388. [[CrossRef](#)]
15. Chang, F.J.; Chen, P.A.; Lu, Y.R.; Huang, E.; Chang, K.Y. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* **2014**, *517*, 836–846. [[CrossRef](#)]
16. Chang, F.J.; Huang, C.W.; Cheng, S.T.; Chang, L.C. Conservation of groundwater from over-exploitation—Scientific analyses for groundwater resources management. *Sci. Total Environ.* **2017**, *598*, 828–838. [[CrossRef](#)] [[PubMed](#)]
17. Chang, F.J.; Tsai, W.P.; Chen, H.K.; Tam, R.S.W.; Herricks, E.E. A self-organizing radial basis network for estimating riverine fish diversity. *J. Hydrol.* **2013**, *476*, 280–289. [[CrossRef](#)]
18. Ghorbani, M.A.; Zadeh, H.A.; Isazadeh, M.; Terzi, O. A comparative study of artificial neural network (MLP, RBF) and support vector machine models for river flow prediction. *Environ. Earth Sci.* **2016**, *75*, 476. [[CrossRef](#)]
19. Ashrafi, M.; Chua, L.H.C.; Quek, C.; Qin, X. A fully-online Neuro-Fuzzy model for flow forecasting in basins with limited data. *J. Hydrol.* **2017**, *545*, 424–435. [[CrossRef](#)]
20. Lohani, A.K.; Goel, N.K.; Bhatia, K.K.S. Improving real time flood forecasting using fuzzy inference system. *J. Hydrol.* **2014**, *509*, 25–41. [[CrossRef](#)]
21. Nguyen, P.K.T.; Chua, L.H.C.; Talei, A.; Chai, Q.H. Water level forecasting using neuro-fuzzy models with local learning. *Neural Comput. Appl.* **2018**, *30*, 1877–1887. [[CrossRef](#)]
22. Yaseen, Z.M.; Jaafar, O.; Deo, R.C.; Kisi, O.; Adamowski, J.; Quilty, J.; El-Shafie, A. Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq. *J. Hydrol.* **2016**, *542*, 603–614. [[CrossRef](#)]
23. Zhou, Y.; Guo, S.; Chang, F.J.; Liu, P.; Chen, A.B. Methodology that improves water utilization and hydropower generation without increasing flood risk in mega cascade reservoirs. *Energy* **2018**, *143*, 785–796. [[CrossRef](#)]
24. Chang, F.J.; Chiang, Y.M.; Ho, Y.H. Multi-step-ahead flood forecasts by neuro-fuzzy networks with effective rainfall-runoff patterns. *J. Flood Risk Manag.* **2015**, *8*, 224–236. [[CrossRef](#)]
25. Chang, F.J.; Tsai, M.J. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *J. Hydrol.* **2016**, *535*, 256–269. [[CrossRef](#)]
26. Zhong, Y.; Guo, S.; Ba, H.; Xiong, F.; Chang, F.J.; Lin, K. Evaluation of the BMA probabilistic inflow forecasts using TIGGE numeric precipitation predictions based on artificial neural network. *Hydrol. Res.* **2018**. [[CrossRef](#)]
27. Zhou, J.; Peng, T.; Zhang, C.; Sun, N. Data Pre-Analysis and Ensemble of Various Artificial Neural Networks for Monthly Streamflow Forecasting. *Water* **2018**, *10*, 628. [[CrossRef](#)]
28. Seng Mah, D.Y.; Lai, S.H.; Chan, R.A.; Putuhena, F.J. Investigative modelling of the flood bypass channel in Kuching, Sarawak, by assessing its impact on the inundations of Kuching-Batu Kawa-Bau Expressway. *Struct. Infrastruct. Eng.* **2012**, *8*, 705–714. [[CrossRef](#)]
29. Benito, G.; Lang, M.; Barriendos, M.; Llasat, M.C.; Frances, F.; Ouarda, T.; Thorndycraft, V.R.; Enzel, Y.; Bardossy, A.; Coeur, D.; et al. Use of systematic, palaeoflood and historical data for the improvement of flood risk estimation. Review of scientific methods. *Nat. Hazards* **2004**, *31*, 623–643.
30. Wallingford Software. InfoWorks RS Forms Basis for Flood Map of Northern Ireland. User Case Studies. Available online: [http://www.wallingfordsoftware.com/casestudies/fullarticle.asp?ID\protect\\\$\\relax\protect\\begin{group1}\\endgroup\\@over4\\\$870](http://www.wallingfordsoftware.com/casestudies/fullarticle.asp?ID\protect\$\\relax\protect\\begin{group1}\\endgroup\\@over4\$870) (accessed on 28 November 2009).
31. Hassan, A.J. *River and Floodplain Modelling: A Practical Approach*; National Hydraulic Research Institute of Malaysia (NAHRIM): Kuala Lumpur, Malaysia, 2009.

32. Adnan, M.S.; Yuliarahmadila, E.; Norfathiah, C.A.; Kasmin, H.; Rosly, N. *Flood Simulation Using Rainfall-Runoff for Segamat River Basin, Advances in Civil, Architectural, Structural and Constructional Engineering*; Kim, J.S., Ed.; Taylor & Francis Group: London, UK, 2016; ISBN 978-1-138-02849-4.
33. Othman, F.; Amin, M.; Farahain, N.; Mi Fung, L.; Elamin, M.; Eldin, A. Utilizing GIS and Infoworks RS in Modelling the Flooding Events for a Tropical River Basin. *Appl. Mech. Mater.* **2013**, *353–356*, 2281–2285. [[CrossRef](#)]
34. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 59–69. [[CrossRef](#)]
35. Chang, L.C.; Shen, H.Y.; Chang, F.J. Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. *J. Hydrol.* **2014**, *519*, 476–489. [[CrossRef](#)]
36. Chang, F.J.; Chang, L.C.; Huang, C.W.; Kao, I.F. Prediction of monthly regional groundwater levels through hybrid soft-computing techniques. *J. Hydrol.* **2016**, *541*, 965–976. [[CrossRef](#)]
37. Chen, I.T.; Chang, L.C.; Chang, F.J. Exploring the spatio-temporal interrelation between groundwater and surface water by using the self-organizing maps. *J. Hydrol.* **2018**, *556*, 131–142. [[CrossRef](#)]
38. Tsai, W.P.; Huang, S.P.; Cheng, S.T.; Shao, K.T.; Chang, F.J. A data-mining framework for exploring the multi-relation between fish species and water quality through self-organizing map. *Sci. Total Environ.* **2017**, *579*, 474–483. [[CrossRef](#)] [[PubMed](#)]
39. Agarwal, A.; Maheswaran, R.; Kurths, J.; Khosa, R. Wavelet Spectrum and self-organizing maps-based approach for hydrologic regionalization—a case study in the western United States. *Water Res. Manag.* **2016**, *30*, 4399–4413. [[CrossRef](#)]
40. Kalteh, A.M.; Hjorth, P.; Berndtsson, R. Review of the self-organizing map (SOM) approach in water resources: Analysis, modelling and application. *Environ. Model. Softw.* **2008**, *23*, 835–845. [[CrossRef](#)]
41. Leontaritis, I.J.; Billings, S.A. Input–output parametric models for nonlinear systems. *Int. J. Control* **1985**, *41*, 303–344. [[CrossRef](#)]
42. Chen, P.A.; Chang, L.C.; Chang, F.J. Reinforced Recurrent Neural Networks for Multi-Step-Ahead Flood Forecasts. *J. Hydrol.* **2013**, *497*, 71–79. [[CrossRef](#)]
43. Chang, F.J.; Chen, P.A.; Chang, L.C.; Tsai, Y.H. Estimating spatio-temporal dynamics of stream total phosphate concentration by soft computing techniques. *Sci. Total Environ.* **2016**, *562*, 228–236. [[CrossRef](#)] [[PubMed](#)]
44. Chai, L.; Qu, Y.; Liang, S.; Wang, J. Estimating time-series leaf area index based on recurrent nonlinear autoregressive neural networks with exogenous inputs. *Int. J. Remote Sens.* **2012**, *33*, 5712–5731. [[CrossRef](#)]
45. Eugen, D. Prediction of chaotic time series with NARX recurrent dynamic neural networks. In Proceedings of the 9th WSEAS International Conference International Conference Automation and Information, Bucharest, Romania, 24–26 June 2008; pp. 248–253.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Identifying the Sensitivity of Ensemble Streamflow Prediction by Artificial Intelligence

Yen-Ming Chiang <sup>1</sup>, Ruo-Nan Hao <sup>1</sup>, Jian-Quan Zhang <sup>1</sup>, Ying-Tien Lin <sup>2</sup> and Wen-Ping Tsai <sup>3,\*</sup>

<sup>1</sup> Institute of Hydrology and Water Resources, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China; chiangym@zju.edu.cn (Y.-M.C.); rolinhao@zju.edu.cn (R.-N.H.); zhangjianquan@gemdale.com (J.-Q.Z.)

<sup>2</sup> Ocean College, Zhejiang University, Hangzhou 316021, China; kevinlin@zju.edu.cn

<sup>3</sup> Department of Civil and Environmental Engineering, Pennsylvania State University, State College, PA 16801, USA

\* Correspondence: wptsai11@gmail.com

Received: 30 August 2018; Accepted: 26 September 2018; Published: 27 September 2018

**Abstract:** Sustainable water resources management is facing a rigorous challenge due to global climate change. Nowadays, improving streamflow predictions based on uneven precipitation is an important task. The main purpose of this study is to integrate the ensemble technique concept into artificial neural networks for reducing model uncertainty in hourly streamflow predictions. The ensemble streamflow predictions are built following two steps: (1) Generating the ensemble members through disturbance of initial weights, data resampling, and alteration of model structure; (2) consolidating the model outputs through the arithmetic average, stacking, and Bayesian model average. This study investigates various ensemble strategies on two study sites, where the watershed size and hydrological conditions are different. The results help to realize whether the ensemble methods are sensitive to hydrological or physiographical conditions. Additionally, the applicability and availability of the ensemble strategies can be easily evaluated in this study. Among various ensemble strategies, the best ESP is produced by the combination of boosting (data resampling) and Bayesian model average. The results demonstrate that the ensemble neural networks greatly improved the accuracy of streamflow predictions as compared to a single neural network, and the improvement made by the ensemble neural network is about 19–37% and 20–30% in Longquan Creek and Jinhua River watersheds, respectively, for 1–3 h ahead streamflow prediction. Moreover, the results obtained from different ensemble strategies are quite consistent in both watersheds, indicating that the ensemble strategies are insensitive to hydrological and physiographical factors. Finally, the output intervals of ensemble streamflow prediction may also reflect the possible peak flow, which is valuable information for flood prevention.

**Keywords:** ensemble technique; artificial neural networks; uncertainty; streamflow predictions; sensitivity

## 1. Introduction

The frequency and intensity of extreme rainfall events have significantly increased due to climate change in past years. Heavy rainfall is the major cause of flood disasters; therefore, there is an urgent need to construct reliable flood prediction models. Due to special geographical and climatic conditions, the Zhejiang province, China, has always been a flood disaster-prone area. Thus, strategies to effectively deal with flood threats have become a priority. Since the influence of global climate change has become increasingly significant, the former balance of rainfall–runoff mechanisms is failing. The occurrence and intensity of extreme hydrological events are more frequent than those in previous years. To reduce the impact of flood hazards, the development of hydrological prediction models is necessary and urgently required.

Artificial neural networks (ANNs) have been widely used in solving a wide range of hydrological problems, such as rainfall–runoff modeling [1,2], regional flood frequency analysis [3], groundwater modeling [4–7], hydrological time series modeling, and reservoir operation [8,9]. Hydrological prediction models based on ANNs can effectively identify the relationship between the input and output in hydrological systems, which can overcome the weaknesses of the conventional method of parameterized modeling. For complex rainfall–runoff modeling, ANNs can also produce reliable outputs through historical data learning. Thus, ANNs have become popular and are generally applied in streamflow predictions over the past decade to lessen flood-induced damage.

However, the uncertainty of ANNs comes from several factors, such as the selection of input variables, model structures, initial weights, and calibration data [10,11]. One way to reduce the ANNs' uncertainty is to integrate the ensemble technique into ANN models. Research in the field of ensemble streamflow predictions (ESP) has been remarkably increasing in order to avoid model errors due to single deterministic results of hydrological prediction [12,13]. Nowadays, ensemble prediction has developed into multimode, multianalysis prediction techniques, which consider the models' uncertainty at the initial state and from mode architecture [14]. Because of the flexible geometry of ANNs, they have been recognized as feasible models for ensemble techniques [15].

Tiwari and Chatterjee [16] developed hourly water level forecasting models using bootstrap based ANNs (BANN). Their results indicated that BANN-hydrologic forecasting models with confidence bounds can improve their reliability for flood forecasts. Kasiviswanathan et al. [17] constructed a prediction interval for ANN rainfall runoff models based on ensemble simulations, which showed that generated ensembles predict the peak flow with less error, and most of their observed flows fall within the constructed prediction interval. To forecast urban water demand, a new hybrid wavelet-bootstrap-neural network model was built and performed more accurate forecasting than the traditional neural network, bootstrap-based neural networks, ARIMA (autoregressive integrated moving average), and ARIMAX (autoregressive integrated moving average model with exogenous input variables) models [18]. Ensemble neural network models have also been successfully applied in potential evapotranspiration prediction [19], probabilistic prediction of local precipitation [20], and short-term forecasting of groundwater levels [21].

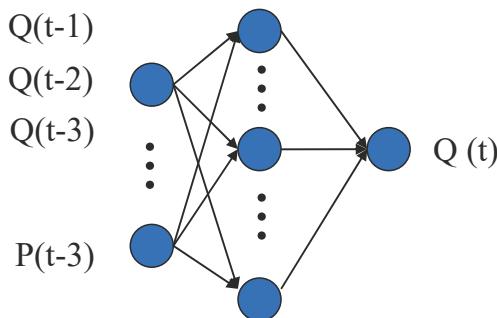
While many studies have applied ensemble techniques to the hydrologic field, there is still a shortage of studies about the sensitivity of ESP. In this paper, the main objective is to integrate the ensemble technique concept into ANNs, hereafter termed ensemble neural networks (ENNs), for reducing the uncertainty in streamflow predictions. ENNs are then applied to two watersheds with different area and hydrological conditions to discuss the sensitivity on ESP. Four methods are used for generating the ensemble members, and three methods are selected to combine the outputs of ensemble members. A total of twelve ensemble strategies are built separately in two different watersheds to validate if the best ESP is consistent and if the best ensemble combination is sensitive to hydrological and physiographical changes. The methodologies of the artificial neural network and two resampling techniques, stacking average, and Bayesian model average, are briefly described in the following section. The study area and hydrological data are provided in Section 3. Section 4 shows the results and comparison of twelve ESP models and the sensitivity analysis of two watersheds. Finally, the conclusions are given in Section 5.

## 2. Methodology

This study aims at integrating the ensemble technique concept into ANNs for constructing accurate ensemble streamflow prediction models and identifying both spatial and hydrological sensitivity of ensemble strategies at two distinct watersheds. The related methods are presented as follow.

## 2.1. Back Propagation Neural Network

The basic concept of artificial neural networks is to simulate the information processing system of biological neural networks by imitating the human nervous system with computer hardware and software. ANNs are composed of many nonlinear operation units, neurons, and links located between the arithmetic units, which usually compute in parallel and dispersedly. The ANNs can be trained through information (data) importing. In this study, the back-propagation neural network (BPNN) [22] is used to construct the streamflow prediction model. The model architecture is shown in Figure 1. The hyperbolic tangent sigmoid transfer function ('tansig' in Matlab) and linear transfer function ('purelin' in Matlab) are used as the activation functions in hidden and output layers, respectively. The number of neurons in the hidden layer is four, which was determined by trial and error. The rainfall and streamflow at the current and previous time step are used as input variables, and the predicted streamflow is the model output. The BPNN applies the steepest descent method to adjust the weights for minimizing the output error. In the learning process, the weights are adjusted by an error convergence technique to obtain the desired output for a given input dataset.



**Figure 1.** The architecture of the streamflow prediction model.

## 2.2. Ensemble Neural Network

The ENN is introduced by integrating the ensemble technique concept into neural networks. The principle of the ensemble method is to construct several specific groups with different model outputs (i.e., a collection of members) to predict a certain target (streamflow in this study), and the difference of each model output provides the probability distribution information of the prediction target. As mentioned above, previous research on artificial neural networks showed that the uncertainty can be classified into three parts: Uncertainties of data, uncertainties of initial values, and uncertainties of the model structure (including the parameters of the model). The ensemble technique has been developed to consider the uncertainties of several sources to avoid network error existing in a single predicted result.

In general, the construction of ENNs can be divided into two steps: Generating ensemble members and integrating the outputs obtained from ensemble members. Methods to generate ensemble members in this study include the disturbance of the initial value, the resampling of the training dataset (Bagging resampling and Boosting resampling), and the alteration of model structure (number of neurons in the hidden layer). The methods selected for combining the outputs of ensemble members include arithmetic averaging, the Bayesian model averaging [15,23], and stack averaging. Another important issue is related to the number of ensemble members. According to Chiang et al. [24], the suggested number of ensemble members used for hydrological forecasting is twenty, which is based on a compromise between output accuracy and computational time. Their recommendation holds for different model types and model structures (i.e., conceptual models and neural networks). Thus, twenty ensemble members were used in this study.

### 2.3. Generating Ensemble Members

As mentioned, the model uncertainty mainly comes from initial values, model structures, and data. Thus, ensemble member generation focuses on reducing these uncertainties. First, a single neural network (SNN), which only uses a calibrated single back-propagation neural network, is generally given random initial values when calibrating the model structure and model parameters. However, ENN in this study starts from a plurality of random initial weights, computes the local optimum value, and extracts useful information to increase the probability of accurate predictions. Subjected to the influence of random initial values, the results obtained may vary in each calibration. Therefore, each network model is trained several times to minimize error of the objective function, which can be regarded as a local optimization. This procedure is repeated 20 times to obtain 20 ensemble members with different initial weights.

Uncertainty from the ANN model structure mainly comes from the number of hidden neurons, since the input and output dimension was fixed in this study. Because the number of ensemble member is 20, the number of hidden neurons from 1 to 20 is assigned to the 20 ensemble members in sequence by using the model structure alternating strategy (ENN4, ENN8, and ENN 12 in Table 1). For example, the hidden neuron for ensemble member 1 is 1, the hidden neuron for ensemble member 2 is 2, and so on. The remaining two strategies of generating members used the best number of hidden neuron, which is the same as the single BPNN (four neurons).

As for the uncertainty of training data, a common method to eliminate its influence is the resampling technique, in which the samples are selected from the original amount of data according to certain rules for enhancing the amount of training sample. The resampling methods applied in this study are the bagging resampling algorithm and boosting resampling algorithm.

The bagging method is proposed for obtaining an aggregated predictor from multiple generated datasets of individual predictors [25]. The assumption of this method is that, given a standard training dataset T of size N, the distribution probability of each element of the training data is uniform, that is, 1/N. Then, the training dataset of a member network, TB, is generated by sampling by replacing N times from the original training dataset T using these probabilities. This process is repeated, and each member of a neural network is generated with a different random sampling, assigned from the original training set.

The boosting algorithm is a method for reducing bias and variance in machine learning and can improve model performance by producing a series of predictors trained with a different distribution of the original training data. The algorithm trains the first member of the neural network with the original training set, and the training dataset of a new member of the neural networks is assigned based on the performance of the previous member of the ensemble. The learning processes in which predicted values produced by the previous member differ significantly from their observations are adjusted with higher probability of being sampled. In this case, these data will have a higher chance to exist in the new training dataset than those correctly predicted, and therefore different members of ensemble are specialized in different parts of the observation space. There are many boosting algorithms, and the procedure of the second version of ADABBoost was used in this study [26].

### 2.4. Integrating Ensemble Members

#### 2.4.1. Arithmetic Averaging

Arithmetic averaging is the simplest averaging method and a popular method for the ensemble technique to combine the models' outputs. Generally, combination using single averaging is defined as:

$$\bar{y} = \frac{1}{K} \sum_{k=1}^K y_i^k, i = 1, 2, \dots, N \quad (1)$$

where K represents the number of ensemble members,  $y$  represents output, and N represents the total number of data points.

#### 2.4.2. Stack Averaging

In general, stacking is not a specific algorithm but a generic name [27]. It means that, when training on part of the training dataset, the performance of the learning machine on the training dataset was not part of the training set for that particular machine giving additional information [28]. The main procedure of stacking is to combine the networks by tuning their weights over the feature space. The outputs obtained from a set of level 0 generalizers (ensemble members) are fed to level 1 generalizer, which is trained to produce appropriate output. The stacking algorithm was developed by Breiman [29], who suggested minimizing the following function:

$$W = \sum_{i=1}^m [x_i - \sum_{k=1}^N c_k \hat{x}_i^k], c_k > 0 \quad (2)$$

The stacked average produces estimates for the coefficients  $c_1, c_2, \dots, c_k$ , which are used to construct the ensemble prediction:

$$\hat{x}_i = \sum_{k=1}^N c_k \hat{x}_i^k, i = 1, 2, \dots, m \quad (3)$$

Equation (3) minimizes squared absolute differences between observations and predictions. This process could be dominated by those patterns with large errors when it is used to calculate the coefficients. A better choice, as adopted in this study, is to minimize the squared relative difference:

$$H = \sum_{i=1}^m \left( \frac{x_i - \sum_{k=1}^N c_k \hat{x}_i^k}{x_i} \right)^2, c_k > 0 \quad (4)$$

#### 2.4.3. Bayesian Model Average

The Bayesian model average (BMA) is capable of obtaining reliable overall predictive values through calculating different weights for all selected models [30–32]. The probability density function of prediction  $y$  based on BMA is as follows:

$$p(y|D) = \sum_{k=1}^K p(f_k|D) \cdot p_k(y|f_k, D) \quad (5)$$

where  $p(f_k|D)$  represents the posterior probability of the  $k$ -th neural network model prediction trained by observed data  $D$ . In fact,  $p(f_k|D)$  is equal to the  $k$ -th model  $f_k$  corresponding to weights  $w_k$ , which is larger when the model performance is better, and  $\sum_{k=1}^K w_k = 1$ ,  $w_k > 0$ .  $p_k(y|f_k, D)$  represents the posterior distribution of prediction  $y$  in neural network model  $f_k$  and data  $D$ . The posterior distribution of the mean and variance BMA analogic variables can be expressed as:

$$E(y|D) = \sum_{k=1}^K p(f_k|D) \cdot E[p_k(y|f_k, D)] = \sum_{k=1}^K w_k f_k \quad (6)$$

$$Var(y|D) = \sum_{k=1}^K w_k \left( f_k - \sum_{k=1}^K w_k f_k \right)^2 + \sum_{k=1}^K w_k \sigma_k^2 \quad (7)$$

where  $\sigma_k^2$  is the variance of analogic variables under the conditions of observed data  $D$  and model  $f_k$ . Essentially, BMA is the weight of the  $k$ -th neural network model's weighted average.

The variance of analogic variables includes error between models and within models. In Equation (7),  $\sum_{k=1}^K \left( f_k - \sum_{k=1}^K w_k f_k \right)^2$  is the error between models, and  $\sum_{k=1}^K w_k \sigma_k^2$  is the error within models.

### 3. Applications

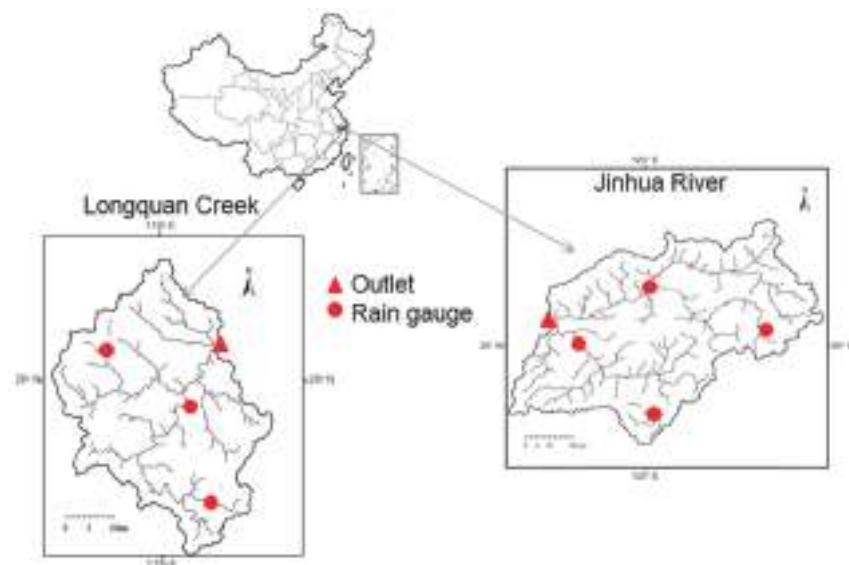
The BPNN was used in building the single and ensemble forecasting models. As for the ensemble scenario, there were a total of twelve ENN models that are implemented for ESP, and these ENN models were derived from the combinations of four generating methods and three combination skills. Detailed information is displayed in Table 1.

**Table 1.** Combinations of ensemble neural networks.

		Integrating Member		
		Arithmetic Average	Bayesian Model Average	Stacking Average
Generating member	Disturbance of initial value	ENN1	ENN5	ENN9
	Bagging resampling skill	ENN2	ENN6	ENN10
	Boosting resampling skill	ENN3	ENN7	ENN11
	Alteration of model structure	ENN4	ENN8	ENN12

#### 3.1. Study Area and Data Description

Longquan Creek is the source of the Oujiang River, which is the second largest river in the Zhejiang province in China (Figure 2). In Figure 2, the triangle represents the watershed outlet and the circles represent the rain gauge locations. Longquan Creek flows to the East sea of China, with a drainage area of  $1440 \text{ km}^2$  and length of 160 km. The watershed receives an annual rainfall of about 1807 mm, and more than 80% of rainfall comes from the monsoon period (from April to June). The hydrological features of Longquan Creek are rapid streamflow and short period of flood peak, due to uneven distribution of rainfall and mountainous topography. This results in the Longquan Creek watershed being a flood-prone area.



**Figure 2.** Study area of Longquan Creek and the Jinhua River watersheds and the locations of gauges.

The Jinhua River is the largest tributary of the Qiantang River, which is the largest river in the Zhejiang province (Figure 2). The Jinhua River flows to the East sea of China, with a drainage area of approximately 6781 km<sup>2</sup> and length of 200 km. Jinhua River watershed is located in a subtropical climate zone. The rainwater of this watershed mainly comes from typhoons. The watershed receives an annual rainfall of approximately 1450 mm. Due to the characteristics of typhoon rainfall (high intensity in a short duration), Jinhua River watershed is also a flood-prone area.

These two watersheds were selected in this study to determine if the ensemble strategies are sensitive to hydrological and physiographical factors. The differences between Longquan Creek and the Jinhua River watershed can be summarized as follow: (1) The watersheds have different shapes and sizes; (2) the rainfall type is monsoon and typhoon rainfall, respectively, and (3) Longquan Creek is located in a mountainous area (upstream) and the Jinhua River is located in a midstream, flat area. Two types of data, hourly streamflow (Q) and average hourly rainfall (P), were used as input variables to build ensemble streamflow predictions in this study. This study used the Pearson correlation coefficient [33] to find the high correlation input variables. The time-dependent variables  $Q(t)$ ,  $Q(t - 1)$ ,  $Q(t - 2)$ , and  $P(t - 3)$ , where  $t$  is the current time, were selected in the Longquan Creek basin, and  $Q(t)$ ,  $Q(t - 1)$ ,  $Q(t - 2)$  and  $P(t - 10)$  were selected in the Jinhua River basin.

A total of 37 flood events occurred in Longquan Creek, and 70 flood events occurred in the Jinhua River during the collection period of 1994 to 2013. Even though the number of events is different, the sample sizes are sufficient to train the neural networks in both watersheds. The arrangement of data in training, validation, and testing phases follows the ratio of 3:1:1. Table 2 shows the statistics of streamflow measurements in three independent datasets. The statistics includes maximum, minimum, mean, and standard deviation (STD) of streamflow.

**Table 2.** The statistics of streamflow in three datasets (m<sup>3</sup>/s).

Statistic	Longquan Creek			Jinhua River		
	Training	Validation	Testing	Training	Validation	Testing
Max.	3040	2430	2640	4200	3810	3640
Min.	13	7	11	2	6	3
Mean	248	216	202	514	575	510
STD	323	318	256	550	628	528

### 3.2. Evaluation Criteria

The coefficient of determination ( $R^2$ ), root mean square error (RMSE), and  $G_{bench}$  index ( $G_{bench}$ ) were used in this study to evaluate the accuracy of a single neural network and ensemble neural network.

#### (i) Coefficient of Determination ( $R^2$ )

$$R^2 = 1 - \frac{\sum_{t=1}^N (Q_{obs}(t+n) - Q_{pre}(t+n))^2}{\sum_{t=1}^N (Q_{obs}(t+n) - \bar{Q}_{obs})^2} \quad (8)$$

where  $Q_{obs}(t+n)$  and  $Q_{pre}(t+n)$  are the observed and predicted flow at time  $t + n$ ,  $\bar{Q}_{obs}(t)$  is the mean of the observed runoff, and  $N$  is the number of data points. The value of  $R^2$  varies between negative infinity and one. Values approaching one indicate higher accuracy in model performance.

(ii) root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^N (Q_{obs}(t+n) - Q_{pre}(t+n))^2}{N}} \quad (9)$$

The merits of models can be significantly reflected through RMSE values when assessing peak values of variables.

(iii)  $G_{bench}$  index ( $G_{bench}$ )

$$G_{bench} = 1 - \frac{\sum_{t=1}^N (Q_{pre}(t+n) - Q_{obs}(t+n))^2}{\sum_{t=1}^N (Q_{obs}(t+n) - Q_{bench}(t))^2} \quad (10)$$

where  $Q_{bench}(t)$  represents the benchmark series of real observed runoff at time  $t$ .  $G_{bench}$  is negative if the model performance is poorer than the benchmark, zero if the model performs as well as the benchmark, and positive if the model is superior. Values closer to one indicate a perfect fit [34].

#### 4. Results and Discussions

In this study, the single BPNN was calibrated using the training dataset, and the validation dataset was applied to check the overfitting issue. Then, the twelve ensemble strategies were integrated into BPNN models to build the ensemble streamflow predictions. Results obtained from SNN and ENNs in both watersheds are described below, as well as the comparison of model accuracy and the sensitivity of ensemble strategies.

##### 4.1. Comparison of a Single Neural Network and Ensemble Neural Network

Table 3 shows the test results of forecasted streamflow for 1–3 h lead time in Longquan Creek watershed by the SNN and twelve ENN models. In general, the results produced by all ensemble models are better than the single network model. Among various ensemble strategies, the combination of boosting and BMA (ENN7) provided about 19–37% improvement in terms of RMSE at different lead times compared to the single model. The overall performance of ENN7 was better than other ensemble strategies for 1–3 h ahead streamflow predictions. Compared to other ENN models, the ENN7 model has a higher  $R^2$ , lower RMSE, and higher  $G_{bench}$ , indicating that the combination of the boosting algorithm and Bayesian model average is more reliable for streamflow predictions. Additionally, according to the comparison of the evaluation criteria (Table 3), it can be seen that the single artificial neural network was capable of producing accurate streamflow predictions with the coefficient of determination ( $R^2$ ) being higher than 0.9 for 1–3 h ahead streamflow predictions. In addition, the use of the ensemble technique effectively increased the output accuracy, which means the integration of the ensemble technique and ANN provides a better option for hydrological predictions.

Table 4 lists the results of the testing dataset obtained from the SNN and twelve ENN models in the Jinhua River watershed. Similar to those by the Longquan Creek watershed, the results produced by ENN models are better than those of the single model. Among various ensemble strategies, the ENN7 model still provided the best performance compared to other ENN models in terms of higher  $R^2$ , lower RMSE, and higher  $G_{bench}$  values. Even though the performance of all ENN models is similar at a lead time of one hour, the results have significant difference as the lead time increases. Compared to the SNN, the improvement made by the ENN7 model is about 20–30% for 1–3 h ahead streamflow predictions in terms of RMSE. The results also demonstrate that the combination of the boosting algorithm and Bayesian model average had a better predictive capability for long-term streamflow predictions. Figures 3 and 4 show the scatterplot of observations and predictions produced by the SNN and ENN7 models in the Longquan Creek and Jinhua River watershed, respectively. It is obvious that

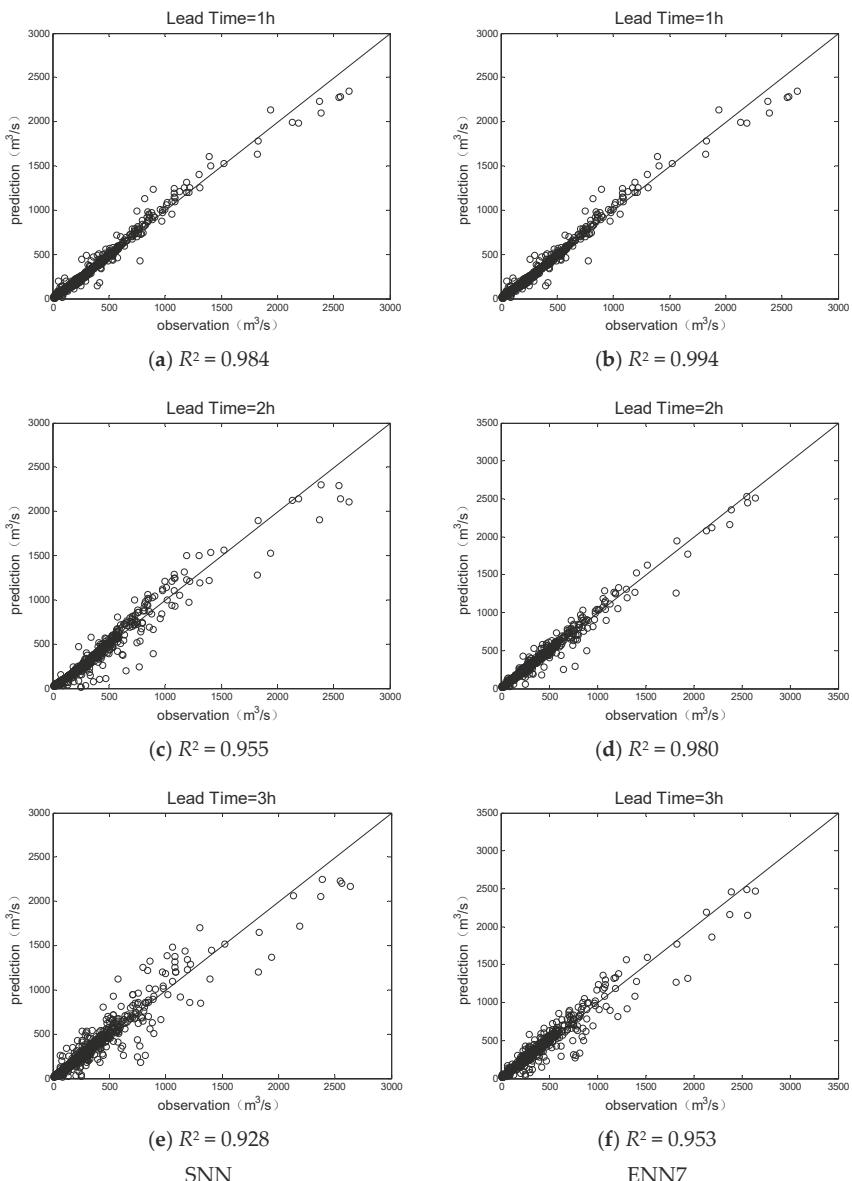
the performance obtained from the ENN7 model is much better than that of the SNN model in both watersheds. According to Tables 3 and 4 and Figures 3 and 4, an important result can be found, which is that the best ensemble strategy (the combination of member generalization and member integration) is neither sensitive to hydrological nor physiographical conditions in terms of streamflow prediction. Therefore, the boosting resampling algorithm is suggested for generating ensemble members and the Bayesian model average is recommended for integrating the outputs of ensemble members.

**Table 3.** Testing results obtained from the single neural network (SNN) and 12 ensemble neural networks (ENN)s in the Longquan Creek watershed.

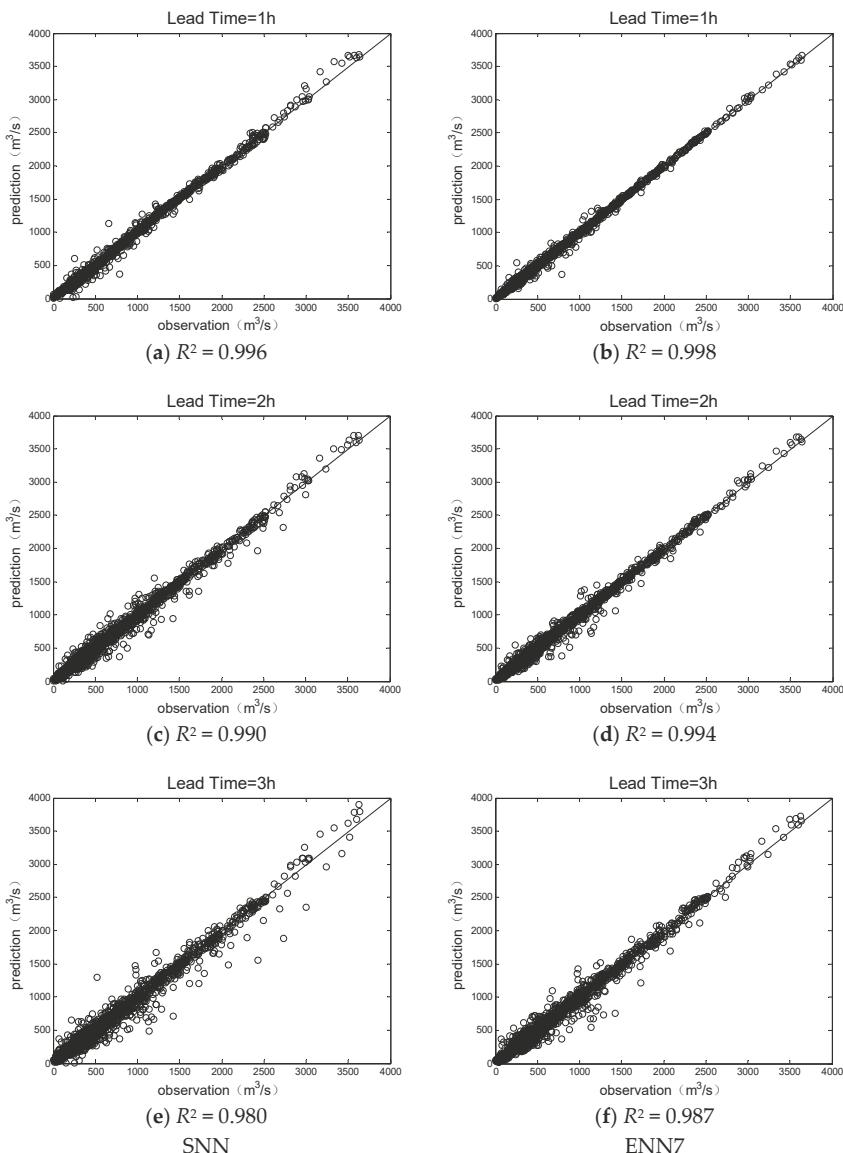
Criteria	$R^2$			RMSE			$G_{bench}$			
	Lead Time	1 h	2 h	3 h	1 h	2 h	3 h	1 h	2 h	3 h
SNN		0.984	0.955	0.928	32.2	54.3	69.0	0.375	0.489	0.587
ENN1		0.993	0.976	0.945	21.9	39.4	60.0	0.712	0.731	0.688
ENN2		0.993	0.977	0.947	21.6	38.8	59.3	0.720	0.740	0.696
ENN3		0.993	0.976	0.945	21.6	39.8	60.3	0.720	0.726	0.685
ENN4		0.993	0.978	0.944	21.6	38.2	60.7	0.720	0.746	0.680
ENN5		0.993	0.976	0.946	21.7	39.5	59.8	0.718	0.730	0.690
ENN6		0.993	0.979	0.950	21.4	37.4	57.5	0.726	0.757	0.713
ENN7		0.994	0.980	0.953	20.2	36.0	55.6	0.754	0.775	0.732
ENN8		0.993	0.975	0.936	21.8	40.2	65.3	0.715	0.719	0.631
ENN9		0.993	0.976	0.943	21.6	39.5	61.3	0.720	0.730	0.674
ENN10		0.991	0.976	0.950	23.8	40.0	57.7	0.659	0.723	0.711
ENN11		0.990	0.979	0.949	25.8	37.3	57.8	0.600	0.759	0.711
ENN12		0.993	0.971	0.928	22.1	43.8	69.0	0.706	0.668	0.587

**Table 4.** Testing results obtained from the SNN and 12 ENNs in the Jinhua River watershed.

Criteria	$R^2$			RMSE			$G_{bench}$			
	Lead Time	1 h	2 h	3 h	1 h	2 h	3 h	1 h	2 h	3 h
SNN		0.996	0.990	0.980	32.3	53.1	75.5	0.827	0.776	0.730
ENN1		0.998	0.993	0.986	23.0	43.3	62.1	0.913	0.851	0.817
ENN2		0.998	0.994	0.987	23.3	42.1	61.1	0.910	0.859	0.823
ENN3		0.998	0.993	0.986	25.9	44.2	62.1	0.888	0.845	0.817
ENN4		0.998	0.992	0.986	23.3	46.0	63.8	0.910	0.832	0.807
ENN5		0.998	0.993	0.987	23.1	42.9	61.8	0.912	0.854	0.819
ENN6		0.998	0.994	0.987	23.5	42.0	60.8	0.909	0.860	0.825
ENN7		0.998	0.994	0.987	22.6	41.8	60.3	0.915	0.862	0.828
ENN8		0.998	0.993	0.985	23.2	44.5	64.0	0.911	0.843	0.806
ENN9		0.998	0.993	0.986	22.7	43.0	62.9	0.915	0.853	0.813
ENN10		0.998	0.994	0.986	23.1	42.5	62.1	0.912	0.857	0.817
ENN11		0.998	0.994	0.987	23.2	42.0	61.1	0.910	0.860	0.823
ENN12		0.998	0.993	0.985	24.2	45.4	64.7	0.903	0.836	0.802



**Figure 3.** Scatterplots of observations and predictions produced by the SNN and ENN7 models in the Longquan Creek watershed.



**Figure 4.** Scatterplots of observations and predictions produced by the SNN and ENN7 models in the Jinhua River watershed.

#### 4.2. Peak Flow Prediction

The most important mission is to accurately predict peak flow. Compared to the deterministic prediction from the single model, the ensemble models provide the probabilistic outputs to reduce the uncertainty of model predictions. Figures 5 and 6 illustrate the comparison between SNN and ENN7 models for the largest peak flow during the testing phase in the Longquan Creek and Jinhua River watershed, respectively. In Figures 5 and 6, the circles represent the actual streamflow, the grey area represents the predictive interval consisting of twenty ensemble members, and the black line represents the model prediction. Generally, both SNN and ENN7 models produced reliable predictions for the lead time of one hour in both watersheds. However, as the lead time increases, the predictive hydrograph produced by SNN has a significant time-lag problem, which may result in the failure of flood warning or flood prevention. The predictive hydrograph obtained from ENN7 has much better predictions, and the time-lag problem is insignificant. Furthermore, the SNN underestimated the streamflow in the rising limb and overestimated the streamflow in the recession limb for 2 h and 3 h lead time in both watersheds. On the other hand, the outputs of ENN7 fit the observations well and the predictive interval produced by twenty ensemble members covers almost the whole of the actual streamflow, indicating the ENN7 model maintained robust predictive capability for 2 h and 3 h ahead streamflow prediction. In Figures 5 and 6, it is seen that most of the observed peak flow is covered by the predictive interval (gray area). In the other words, this demonstrates that ENN can effectively reduce the quantitative uncertainty of hydrologic models [35,36].

Based on the presented results, it is found that the model accuracy in the Jinhua River watershed is slightly better than that in the Longquan Creek watershed. This is mainly because the Longquan Creek watershed is located upstream, where the flow velocity is much higher than that in midstream and downstream. Thus, further analysis of peak flow prediction must be discussed. Table 5 displays the peak flow predictions obtained from the ENN7 model on the first three largest flood events from the testing dataset in both watersheds. The relative error of model predictions in the Longquan Creek watershed and the Jinhua River watershed are within 10% and 5%, respectively, for 1–3 h ahead streamflow prediction, suggesting that the ENN model was able to produce reliable peak flow predictions.

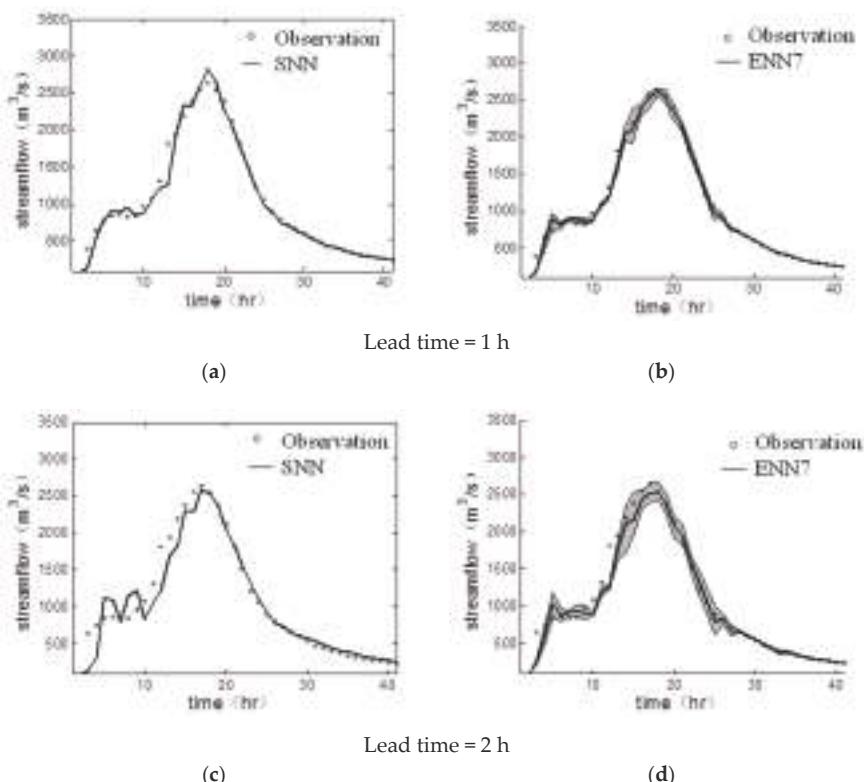
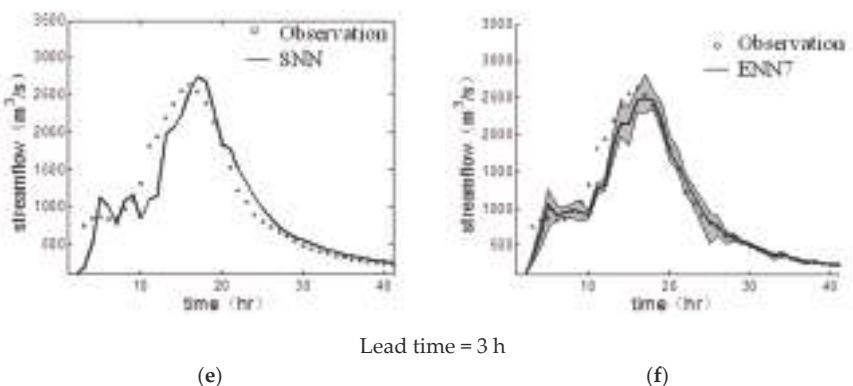


Figure 5. Cont.



**Figure 5.** Comparison between SNN and ENN7 models for the largest peak flow of the testing phase in the Longquan Creek watershed. (a) SNN, lead time = 1 h; (b) ENN7, lead time = 1 h; (c) SNN, lead time = 2 h; (d) ENN7, lead time = 2 h; (e) SNN, lead time = 3 h; (f) ENN7, lead time = 3 h.

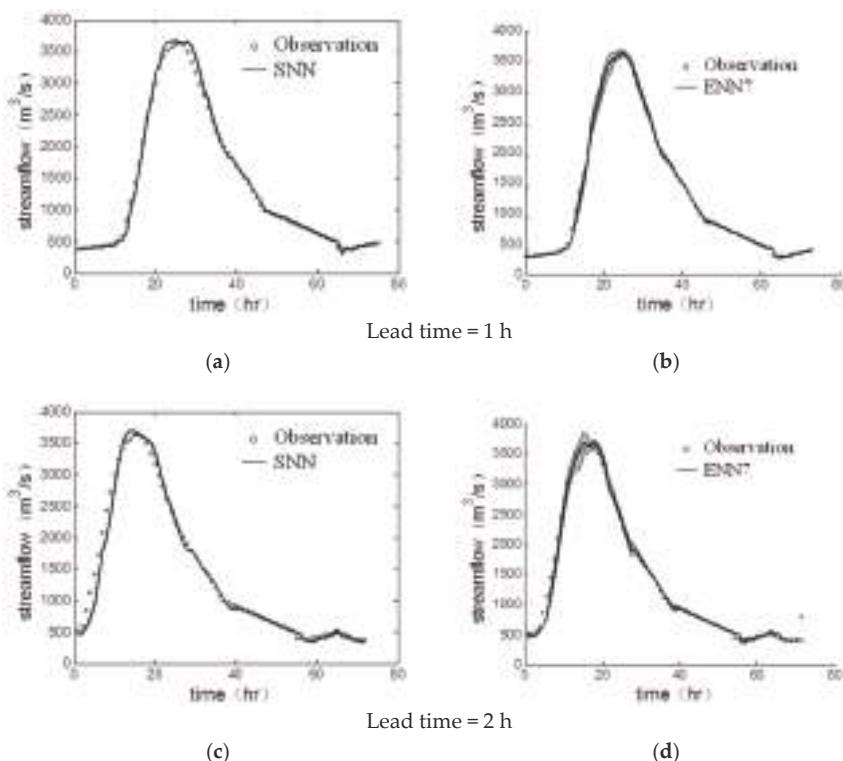
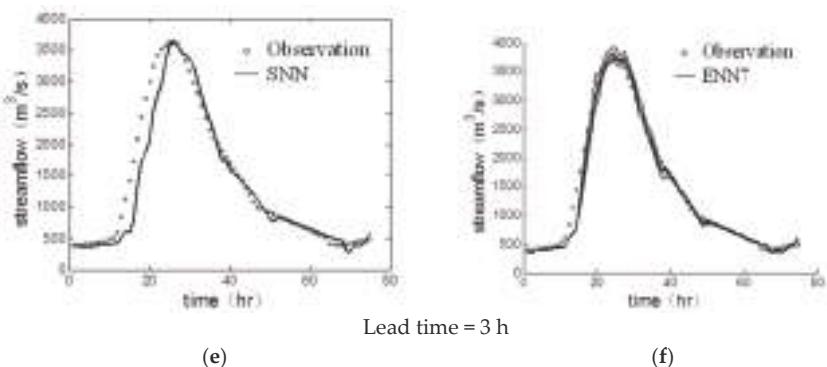


Figure 6. Cont.



**Figure 6.** Comparison between SNN and ENN7 models for the largest peak flow of the testing phase in the Jinhua River watershed. (a) SNN, lead time = 1 h; (b) ENN7, lead time = 1 h; (c) SNN, lead time = 2 h; (d) ENN7, lead time = 2 h; (e) SNN, lead time = 3 h; (f) ENN7, lead time = 3 h.

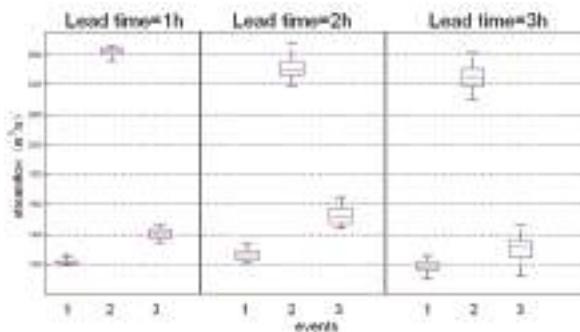
**Table 5.** Peak flow prediction produced by ENN7 in both watersheds.

ENN7	Lead Time = 1 h		Lead Time = 2 h		Lead Time = 3 h		
	Qp	Qp' (m <sup>3</sup> /s)	Error (%)	Qp' (m <sup>3</sup> /s)	Error (%)	Qp' (m <sup>3</sup> /s)	Error (%)
Longquan Creek Basin							
Event 1	1191	1215	2	1261	6	1181	-0.8
Event 2	2640	2618	-0.8	2510	-5	2466	-6.5
Event 3	1410	1403	-0.5	1524	8	1280	-9
Jinhua River Basin							
Event 1	2520	2523	0.1	2507	-0.5	2508	-0.5
Event 2	3040	3062	0.7	3045	0.2	3165	4.1
Event 3	3640	3665	0.7	3611	-0.8	3731	2.5

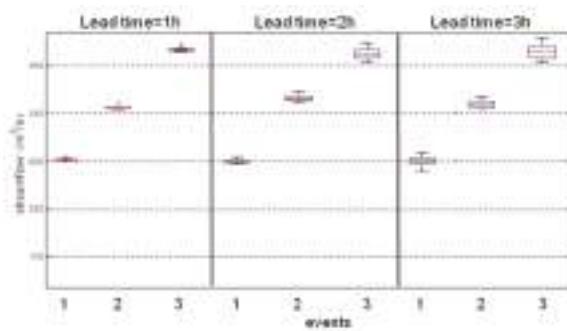
Qp: Actual peak flow; Qp': Forecasted peak flow.

#### 4.3. Sensitivity of Ensemble Neural Networks

To understand whether the ensemble models are sensitive to hydrological and physiographical conditions, Figures 7 and 8 show the peak flow predictions of the three largest events and the RMSE of the testing dataset at the two watersheds, respectively. It is clear from the boxplot (Figure 7) that the peak flow predictions of twelve ENN models are quite stable and consistent for 1–3 h ahead prediction. Figure 8 indicates the overall performance of twelve ENN models in the testing phase. In general, longer lead time may cause larger bias of model output (larger RMSE value when lead time = 3 h, see Figure 8). However, for the streamflow prediction, the results still show a consistent trend of the twelve ENN models in both watersheds for 1–3 h ahead prediction, in which the ENN7 has the lowest RMSE in both the Longquan Creek and Jinhua River watershed. This result demonstrates that the ESP with the combination of the boosting resampling algorithm (generating member) and Bayesian model average (integrating member) was better than others. Figure 8 also shows the trends of RMSE values of the 12 ENNs in both watersheds. There are three ensemble combinations with different trends of RMSE values in both watersheds (ENN3 and ENN11 when lead time = 1 h, ENN4 when lead time = 2 h). The results indicate that RMSE values depend on ensemble combinations. However, the figure demonstrates that there are similar trends (9 of 12 are similar) in both watersheds (lower sensitivity of ensemble neural networks). In summary, the results displayed in Figures 7 and 8 also indicate the ENN models not only present a higher accuracy of predictive capability, but also reveal their lower sensitivity in ESP.

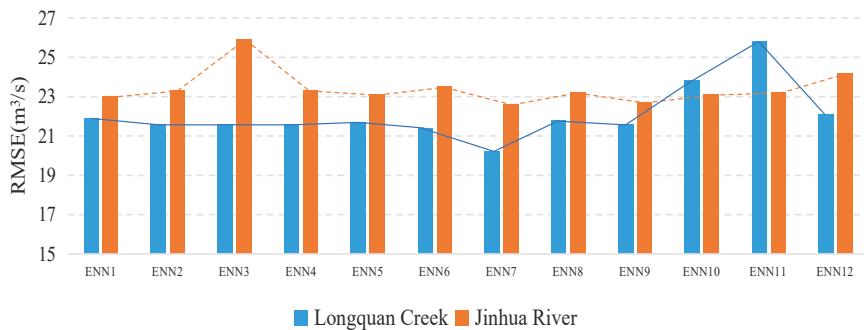


(a) Longquan Creek



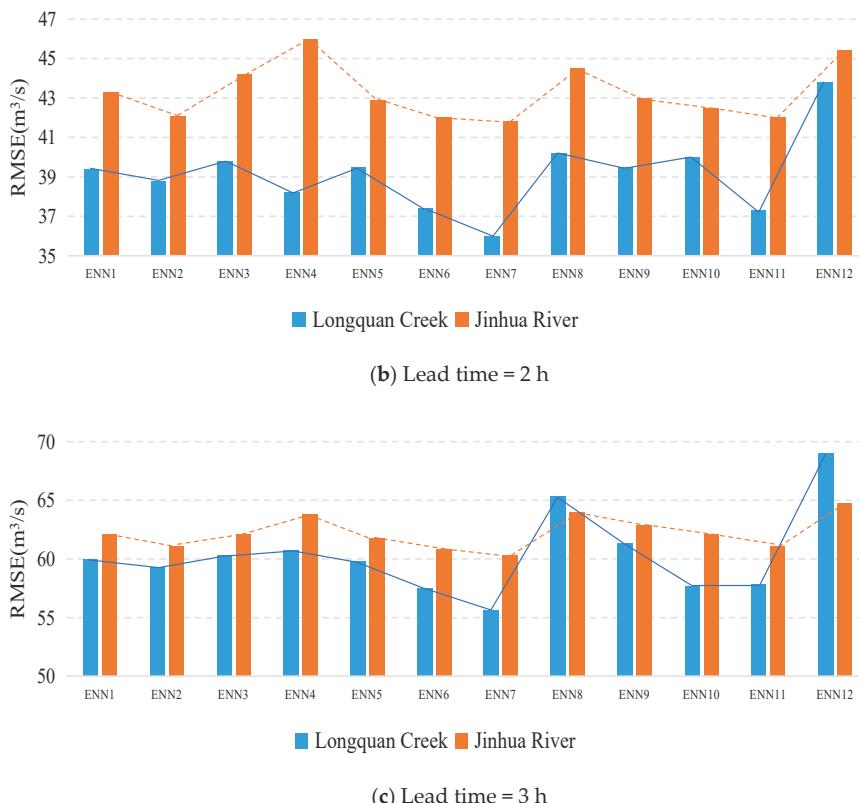
(b) Jinhua River

**Figure 7.** The boxplot of peak flow prediction on the first three largest events obtained from 12 ENN models (a) Longquan Creek; (b) Jinhua River.



(a) Lead time = 1 h

**Figure 8. Cont.**



**Figure 8.** The comparison of ENN models at both watersheds (a) Lead time = 1 h; (b) Lead time = 2 h; (c) Lead time = 3 h.

## 5. Conclusions

Streamflow prediction is critical for assessing imminent flood risk and evaluating and planning flood mitigation activities. In general, uncertainty and sensitivity are two important considerations in hydrological modeling. The main purpose of this study was to integrate the ensemble technique concept into artificial neural networks to reduce uncertainty and discuss the sensitivity in streamflow predictions. The results show that the ENNs were able to effectively reduce the uncertainty in hydrological modeling, compared to the SNN. Additionally, the best ensemble strategy was identified through both case studies as the combination of boosting resampling and Bayesian model average. The main achievements and innovations of this study are concluded as follows: (1) The ENN models greatly improved the accuracy of streamflow prediction compared to SNN models for 1–3 h ahead prediction in both watersheds. The improvement made by the ENNs is about 20% to 40% in terms of RMSE; (2) the relative error of peak flow predictions in both the Longquan Creek and Jinhua River watershed obtained from the ENN7 model demonstrates that the ensemble model is capable of reflecting the possible maximum flood, which is a valuable reference for flood prevention; (3) the best ensemble strategy integrated into the ANN-based hydrological models in two study watersheds is the same, indicating that the ensemble strategy has low sensitivity to the hydrological and physiographical factors. In other words, the artificial neural network combined with ensemble technique can be applicable for generating streamflow predictions in different flood-prone areas.

**Author Contributions:** Conceptualization, W.-P.T.; Formal analysis, R.-N.H.; Methodology, J.-Q.Z.; Project administration, Y.-M.C.; Software, R.-N.H.; Supervision, Y.-M.C.; Writing—Original draft, Y.-M.C.; Writing—Review & editing, Y.-T.L.

**Funding:** This research was funded by the (National Key Research and Development Program of China Inter-governmental Cooperation in International Scientific and Technological Innovation) grant number (2016YFE0122100), (Fundamental Research Funds for the Central Universities) grant number (2018QNA4030), and (National Key Research and Development Program of China) grant number (2016YFC0402407).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Chang, F.J.; Tsai, M.J. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *J. Hydrol.* **2016**, *535*, 256–269. [[CrossRef](#)]
- Chokmani, K.; Ouarda, T.; Hamilton, S. Comparison of ice-affected streamflow estimates computed using artificial neural networks and multiple regression techniques. *J. Hydrol.* **2008**, *349*, 383–396. [[CrossRef](#)]
- Shu, C.; Ouarda, T.B.M.J. Flood frequency analysis at ungauged sites using artificial neural networks in canonical correlation analysis physiographic space. *Water Resour. Res.* **2007**, *43*, W07438. [[CrossRef](#)]
- Gong, Y.; Zhang, Y.; Lan, S.; Wang, H. A Comparative Study of Artificial Neural Networks, Support Vector Machines and Adaptive Neuro Fuzzy Inference System for Forecasting Groundwater Levels near Lake Okeechobee, Florida. *Water Resour. Manag.* **2016**, *30*, 375–391. [[CrossRef](#)]
- Maiti, S.; Tiwari, R.K. A comparative study of artificial neural networks, Bayesian neural networks and adaptive neuro-fuzzy inference system in groundwater level prediction. *Environ. Earth Sci.* **2014**, *71*, 3147–3160. [[CrossRef](#)]
- Sreekanth, J.; Datta, B. Stochastic and Robust Multi-Objective Optimal Management of Pumping from Coastal Aquifers Under Parameter Uncertainty. *Water Resour. Manag.* **2014**, *28*, 2005–2019. [[CrossRef](#)]
- Tsai, W.P.; Chiang, Y.M.; Huang, J.L.; Chang, F.J. Exploring the Mechanism of Surface and Ground Water through Data-Driven Techniques with Sensitivity Analysis for Water Resources Management. *Water Resour. Manag.* **2016**, *30*, 4789–4806. [[CrossRef](#)]
- Tsai, W.P.; Chang, F.J.; Chang, L.C.; Herricks, E.E. AI techniques for optimizing multi-objective reservoir operation upon human and riverine ecosystem demands. *J. Hydrol.* **2015**, *530*, 634–644. [[CrossRef](#)]
- Yin, X.A.; Yang, Z.F.; Petts, G.E.; Kondolf, G.M. A reservoir operating method for riverine ecosystem protection, reservoir sedimentation control and water supply. *J. Hydrol.* **2014**, *512*, 379–387. [[CrossRef](#)]
- Deolia, V.K.; Purwar, S.; Sharma, T.N. Stabilization of Unknown Nonlinear Discrete-Time Delay Systems Based on Neural Network. *Intell. Control Autom.* **2012**, *3*, 337–345. [[CrossRef](#)]
- Goyal, V.; Deolia, V.K.; Sharma, T.N. Neural network based sliding mode control for uncertain discrete-time nonlinear systems with time-varying delay. *Int. J. Comput. Intell. Res.* **2016**, *12*, 125–138.
- Dechant, C.M.; Moradkhani, H. Improving the characterization of initial condition for ensemble streamflow prediction using data assimilation. *Hydrol. Earth Syst. Sci.* **2011**, *15*, 3399–3410. [[CrossRef](#)]
- Faber, B.A.; Stedinger, J.R. Reservoir optimization using sampling SDP with ensemble streamflow prediction (ESP) forecasts. *J. Hydrol.* **2001**, *249*, 113–133. [[CrossRef](#)]
- Hao, Z.; Hao, F.; Singh, V.P. A general framework for multivariate multi-index drought prediction based on Multivariate Ensemble Streamflow Prediction (MESP). *J. Hydrol.* **2016**, *539*, 1–10. [[CrossRef](#)]
- Araghinejad, S.; Azmi, M.; Khalghi, M. Application of artificial neural network ensembles in probabilistic hydrological forecasting. *J. Hydrol.* **2011**, *407*, 94–104. [[CrossRef](#)]
- Tiwari, M.K.; Chatterjee, C. Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (BANNs). *J. Hydrol.* **2010**, *382*, 20–33. [[CrossRef](#)]
- Kasiviswanathan, K.S.; Cibin, R.; Sudheer, K.P.; Chaubey, I. Constructing prediction interval for artificial neural network rainfall runoff models based on ensemble simulations. *J. Hydrol.* **2013**, *499*, 275–288. [[CrossRef](#)]
- Tiwari, M.K.; Adamowski, J. Urban water demand forecasting and uncertainty assessment using ensemble wavelet-bootstrap-neural network models. *Water Resour. Res.* **2013**, *49*, 6486–6507. [[CrossRef](#)]
- El-Shafie, A.; Najah, A.; Alsulami, H.M.; Jahanbani, H. Optimized Neural Network Prediction Model for Potential Evapotranspiration Utilizing Ensemble Procedure. *Water Resour. Manag.* **2014**, *28*, 947–967. [[CrossRef](#)]

20. Ohba, M.; Kadokura, S.; Yoshida, Y.; Nohara, D.; Toyoda, Y. Downscaling medium-range ensemble forecasts using a neural network approach. *Proc. Int. Assoc. Hydrol. Sci.* **2015**, *369*, 7–11. [[CrossRef](#)]
21. Khalil, B.; Broda, S.; Adamowski, J.; Ozga-Zielinski, B.; Donohoe, A. Short-term forecasting of groundwater levels under conditions of mine-tailings recharge using wavelet ensemble neural network models. *Hydrogeol. J.* **2015**, *23*, 121–141. [[CrossRef](#)]
22. Rumelhart, D.E. *Learning Internal Representations by Error Propagation*; MIT Press: Cambridge, MA, USA, 1986; ISBN 0-262-68053-X.
23. Ouarda, T.B.M.J.; Shu, C. Regional low-flow frequency analysis using single and ensemble artificial neural networks. *Water Resour. Res.* **2009**, *45*, 114–122. [[CrossRef](#)]
24. Chiang, Y.M.; Hao, R.N.; Ho, H.C.; Chang, T.J.; Xu, Y.P. Evaluating the contribution of multi-model combination to streamflow hindcasting by empirical and conceptual models. *Hydrol. Sci. J.* **2017**, *62*, 1456–1468. [[CrossRef](#)]
25. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
26. Freund, Y.; Schapire, R.E. Experiments with a New Boosting Algorithm. In Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
27. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [[CrossRef](#)]
28. Drucker, H. Improving Regressors using Boosting Techniques. In Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA, 8–12 July 1997; pp. 107–115.
29. Breiman, L. Stacked regressions. *Mach. Learn.* **1996**, *24*, 49–64. [[CrossRef](#)]
30. Hoeting, J.A.; Madigan, D.M.; Raftery, A.E.; Volinsky, C.T. Bayesian model averaging: A tutorial (with discussion). *Stat. Sci.* **1999**, *14*, 382–401.
31. Kass, R.E.; Raftery, A.E. Bayes factors. *J. Am. Stat. Assoc.* **1995**, *90*, 773–795. [[CrossRef](#)]
32. Leamer, E.E. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*; Wiley: New York, NY, USA, 1978; p. 370, ISBN 978-0471015208.
33. Pearson, K. Notes on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond.* **1895**, *58*, 240–242. [[CrossRef](#)]
34. Seibert, J. On the need for benchmarks in hydrological modelling. *Hydrol. Process.* **2001**, *15*, 1063–1064. [[CrossRef](#)]
35. Abbaspour, K.C.; Johnson, A.; van Genuchten, M.T. Estimating uncertain flow and transport parameters using a sequential uncertainty fitting procedure. *Vadose Zone J.* **2004**, *3*, 1340–1352. [[CrossRef](#)]
36. Xiong, L.; Wan, M.; Wei, X.; O'Connor, K.M. Indices for assessing the prediction bounds of hydrological models and application by generalised likelihood uncertainty estimation. *Hydrol. Sci. J.* **2009**, *54*, 852–871. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Flood Forecasting Based on an Improved Extreme Learning Machine Model Combined with the Backtracking Search Optimization Algorithm

Lu Chen <sup>1</sup>, Na Sun <sup>1,\*</sup>, Chao Zhou <sup>2</sup>, Jianzhong Zhou <sup>1</sup>, Yanlai Zhou <sup>3</sup>, Junhong Zhang <sup>4</sup> and Qing Zhou <sup>1</sup>

- <sup>1</sup> School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; chen\_lu@hust.edu.cn (L.C.); jz.zhou@hust.edu.cn (J.Z.); m201773780@hust.edu.cn (Q.Z.)  
<sup>2</sup> Changjiang Institute of Survey, Planning, Design and Research, Wuhan 430010, China; zhouchao2@cjwsjy.com.cn  
<sup>3</sup> Department of Geosciences, University of Oslo, 1047 Oslo, Norway; zyl23bulls@whu.edu.cn  
<sup>4</sup> College of Resource and Environment, South-Central University for Nationalities, Wuhan 430074, China; zjh411891611@163.com  
\* Correspondence: sunna@hust.edu.cn

Received: 6 September 2018; Accepted: 22 September 2018; Published: 29 September 2018

**Abstract:** Flood forecasting plays an important role in flood control and water resources management. Recently, the data-driven models with a simpler model structure and lower data requirement attract much more attentions. An extreme learning machine (ELM) method, as a typical data-driven method, with the advantages of a faster learning process and stronger generalization ability, has been taken as an effective tool for flood forecasting. However, an ELM model may suffer from local minima in some cases because of its random generation of input weights and hidden layer biases, which results in uncertainties in the flood forecasting model. Therefore, we proposed an improved ELM model for short-term flood forecasting, in which an emerging dual population-based algorithm, named backtracking search algorithm (BSA), was applied to optimize the parameters of ELM. Thus, the proposed method is called ELM-BSA. The upper Yangtze River was selected as a case study. Several performance indexes were used to evaluate the efficiency of the proposed ELM-BSA model. Then the proposed model was compared with the currently used general regression neural network (GRNN) and ELM models. Results show that the ELM-BSA can always provide better results than the GRNN and ELM models in both the training and testing periods. All these results suggest that the proposed ELM-BSA model is a promising alternative technique for flood forecasting.

**Keywords:** flood forecasting; extreme learning machine (ELM); backtracking search optimization algorithm (BSA); the upper Yangtze River

## 1. Introduction

Flood forecasting is not only an effective tool to reduce many risks posed by floods on life, property, and infrastructures, but can also provide valuable decision-making information for water resource managers [1–4]. However, due to streamflow affected by human activities and various hydro-meteorological factors, such as rainfall, topography, and surface heterogeneity, the runoff process exhibits highly non-linear, non-stationary, and complexly dynamic behaviors. Therefore, accurate flood forecasting, especially in the short-term (hourly or daily scale), has been recognized as a challenging work in hydrology.

Until now, plenty of hydrological models have been established to realize the flood forecasting [1]. These prediction models can be broadly classified into two kinds, namely physical-based models

(also called knowledge-based models) and data-driven models (DDMs). The first group of models usually imitate the complex behaviors in the hydrologic cycle system by conceptualizing physical processes and basin characteristics, which often depends on detailed information and deep understanding about physical mechanisms of hydrological processes. Additionally, fine modelling of physical-based forecasting models using a full set of mathematic equations for each part in the hydrological cycle (i.e., interception, infiltration, evaporation) can theoretically reflect the real-world hydrological cycle more accurately, but this can lead to many intractable complications, such as the massive parameters to be estimated, the plenty of data requirements, and the expensive computational costs [5–8]. Compared with the physical-based models, the DDMs with a simpler model structure and less demanding data attract much more attention as an alternative forecasting tool in the cases that cannot reach the modelling conditions of physical-based models. Moreover, the rapid developments in computer sciences and some new technologies regarding machine learning, data mining, and optimization algorithms provide new opportunities for the DDMs in the application of various study domains including flood forecasting.

Over the last several decades, various DDMs were developed for flood forecasting, such as the artificial neural networks (ANNs) [1,9–12], adaptive neural-based fuzzy inference systems (ANFIS) [13,14], and support vector machines [15]. Among them, single hidden-layer feedforward neural networks (SLFNs), as the most widely used DDMs, show a strong ability to characterize any nonlinear mapping relationship, and have been taken as effective tools in solving many practical problems, such as flood forecasting [10,16–18], water level forecasting [19,20], and wind speed forecasting [21,22]. Although SLFNs have been successfully applied for modeling hydrological time series, they still suffer from several inherent disadvantages such as a slow learning process, easy plunging into local minima, and an over-fitting problem.

Recently, a novel learning algorithm for SLFN models, called the extreme learning machine (ELM), was developed by Huang et al. [23]. Compared with other typical SLFNs using gradient-based learning (GL) algorithms that learn parameters of a network in an iterative way, ELM is not involves less calculation work, higher learning speed, and stronger generalization ability, but also has no requirements for some parameters, such as terminating condition and learning rate. Considering these features, ELM has been applied as a promising non-linear fitting tool in massive complicated engineering applications [9,21,22]. For example, Yaseen et al. [24] applied the ELM for predicting the monthly streamflow discharge rates in a semi-arid region in Iraq and demonstrated its superiority over support vector regression (SVR) and general regression neural network (GRNN) models. In the same year, Deo and Şahin [20] testified the performance of ELM over conventional ANNs in forecasting mean streamflow water level based on many hydro-meteorological factors. More recently, Zhou et al. [9] developed a GRNN-based ensemble technique (GNE) for monthly streamflow forecasting, in which the results of three famous ANNs, namely radial basis function, ELM, and Elman networks, were fed into a GRNN model as the inputs.

Despite many successful applications of ELM in flood forecasting, it also results in an ill-conditioned problem in some cases because of its random mechanism in generating input weights and hidden layer biases. Therefore, it is necessary to introduce some effective techniques/tools to improve the generalization performance of the single ELM. To date, many endeavors have been made to enhance the stability of the basic ELM. The most famous way is that an evolutionary algorithm was adopted to search the optimal hidden node parameters of ELM. Han et al. [25] proposed a hybrid learning algorithm, in which an improved particle swarm optimization (IPSO) algorithm was applied to adjust the parameters of an ELM. Results showed that the developed IPSO-ELM approach had better generalization performance than the conventional ELM and the other evolutionary ELMs based on a differential evolution algorithm (DE) or PSO algorithm. Recently, a novel dual population-based iterative evolution algorithm, namely backtracking search optimization algorithm (BSA), was proposed in 2013 [26]. Since then, BSA has been used as an effective technique for searching global optimization. Unlike other widely used evolutionary algorithms (EAs), such as PSO, covariance matrix adaptation

evolution strategy (CMAES), artificial bee colony algorithm (ABC), adaptive DE algorithm (JDE), comprehensive learning PSO (CLPSO), and self-adaptive DE algorithm (SADE), BSA has a simpler architecture with only one control parameter, and is insensitive to the initial value of its control parameter. All these features make BSA more effective, adaptive, and faster than other popular EAs. As such, BSA has already been applied to cope with many complex numerical optimization problems as an effective global searching algorithm [26]. However, until now, the capacity of BSA for dealing with the regression problems in the hydrological domain has never been explored.

Therefore, the major objective of this study is to develop a new, improved ELM (ELM-BSA) techniques for daily flood forecasting, which fuses the advantages of ELM and BSA. In the proposed ELM-BSA model, BSA was applied to find the suitable hidden node parameters of ELM, which can further promote the robustness of the standard ELM. The Yangtze River was selected as a case study. The measured daily streamflow data from the Yichang gauging station, the control site of the Three Gorges Reservoir (TGR), was employed to testify the performance of the proposed method. Moreover, two basic DDMs, namely ELM and GRNN models, which are recognized as the most efficient methods for flood forecasting [9,16,27], were selected as benchmark models for comparisons.

The paper is organized as follows. Section 2 introduces the proposed ELM-BSA method for short-term flood forecasting. Section 3 presents a case study of the upper Yangtze River and gives the forecasting results and comparisons with two basic data-driven models. All the conclusions of this study are summarized in Section 4.

## 2. Methodologies

### 2.1. Flood Forecasting Based on the Data-Diven Model

An analytic expression of a flood forecasting model can be defined as:

$$Q(t) = \varphi(Q(t - d_1 + 1), R(t - d_2 + 1), E(t - d_3 + 1)) \quad (1)$$

where  $Q(t)$  is the predicted streamflow at time  $t$ ;  $Q(t - d_1 + 1)$  represents the previous flow up to  $t - d_1 + 1$  time steps;  $R(t - d_2 + 1)$  stands for the antecedent rainfall with  $t - d_2 + 1$  time steps;  $E(t - d_3 + 1)$  is the other relevant factors up to  $t - d_3 + 1$  time steps that have main contributions to the flow at current time  $t$ , such as potential evapotranspiration, temperature, and/or the flow from major control stations in the upper reaches;  $d_i, i = 1, 2, 3$  is the length of time lag for the relevant factors; and  $\varphi(\cdot)$  is a hydrological system transfer function to characterize the complicated nonlinear mapping relationship in a basin between flow and the relevant factors. Two kinds of methods can be used to estimate the  $\varphi(\cdot)$ . The first one is by using physical models (such as Xin'anjiang hydrological model). The second one is by using the data-driven models (i.e., ANN models).

Generally, flood forecasting based on data-driven models can be an alternative method for flood forecasting in some situations, such as when the observed data in the study area are inadequate and/or the potential physical mechanisms of hydrological phenomenon are unknown or only partially understood [8,28]. Moreover, DDMs are easy to establish and can provide acceptable forecasting results with less input data (only rainfall and/or flow data). Considering all these advantages of data-driven models, in this study we developed a new data-driven model named ELM-BSA for flood forecasting. In the new method, ELM, a novel data-driven model, was adopted as a base forecasting module to simulate the hydrological system transfer function  $\varphi(\cdot)$ . Meanwhile, BSA was applied to find the optimal input weights and biases of hidden layer nodes in the ELM to improve the stability of forecasting. The related methods and theories used in the new model, as well as its whole implementation, are presented as follows.

### 2.2. Extreme Learning Machine

An extreme learning machine (ELM) is an emerging fast-learning algorithm for SLFNs that usually has a three-layer structure with one input layer containing  $m$  nodes, one hidden layer containing

$h$  neurons, and a single output layer possessing  $p$  nodes (in flood forecasting,  $p$  is usually set to 1). Usually, the ELM model first randomly selects its input weights and hidden layer biases, and then analytically calculates its output weights using a least squares method instead of iterative adjusting. Therefore, ELM not only possesses the ability of an extremely fast learning speed, but also avoids frequent human intervention, which can provide better performance. These advantages make ELM more and more popular in handling many complex engineering problems.

For a given training sample set  $(\mathbf{X}_j, t_j)$  with  $N$  pairs of observed data, where  $\mathbf{X}_j$  is a multiple-dimensional input vector and  $t_j$  is the target/desired output, the simulated output of ELM can be estimated using:

$$y_j = \sum_{i=1}^h \beta_i g(\omega_i \mathbf{X}_j + b_i), \quad j = 1, 2, \dots, N \quad (2)$$

where  $y_j$  is the output vector of the ELM model using the input vector  $\mathbf{X}_j$ ;  $\beta_i$  denotes the weight vector connecting the  $i$ th hidden neuron to output layer neuron;  $g$  is the activation function for the hidden layer in ELM;  $\omega_i$  are the input weights connecting input layer neurons with the  $i$ th hidden layer neuron; and  $b_i$  and  $g(\omega_i \mathbf{X}_j + b_i)$  are the threshold and output of the  $i$ th hidden node, respectively.

The objective of an ELM is to search for a suitable set of  $\beta$ ,  $\omega$ , and  $b$  to approximate all training sample pairs with zero error:

$$\sum_{j=1}^N \|t_j - y_j\| = \sum_{j=1}^N \|t_j - \sum_{i=1}^h \beta_i g(\omega_i \mathbf{X}_j + b_i)\| = 0 \quad (3a)$$

Equation (3a) can be reorganized to be:

$$H\beta = T \text{ where} \\ H = \begin{bmatrix} g(\omega_1 \mathbf{X}_1 b_1) & g(\omega_2 \mathbf{X}_1 b_2) & \cdots & g(\omega_h \mathbf{X}_1 b_h) \\ g(\omega_1 \mathbf{X}_2 b_1) & g(\omega_2 \mathbf{X}_2 b_2) & \cdots & g(\omega_h \mathbf{X}_2 b_h) \\ \vdots & \vdots & & \vdots \\ g(\omega_1 \mathbf{X}_N b_1) & g(\omega_2 \mathbf{X}_N b_2) & \cdots & g(\omega_h \mathbf{X}_N b_h) \end{bmatrix}_{N \times h} \\ \beta = [\beta_1 \ \beta_2 \ \cdots \ \beta_h]_{h \times 1}^{-1}, \text{ and } T = [t_1 \ t_2 \ \cdots \ t_N]_{N \times 1}^{-1} \quad (3b)$$

where  $H$  is the output matrix of the hidden layer;  $\beta$  is the weights vector connecting the hidden layer nodes with the output layer neurons; and  $T$  represents the target output.

Once the random generation of the input hidden weights and biases of the hidden layer has been completed, ELM analytically calculates the hidden-output weights by searching a minimal norm least square solution of the following linear equation:

$$\|H\hat{\beta} - T\| = \min_{\beta} \|H\beta - T\| \rightarrow 0 \quad (4)$$

The optimal estimated least squares solution of the above equation is:

$$\hat{\beta} = H^{\dagger}T \quad (5)$$

where  $H^{\dagger}$  denotes the Moore–Penrose generalized inverse of the hidden-layer output matrix  $H$ .

### 2.3. Backtracking Search Optimization Algorithm

Inspired by swarm behaviors, i.e., natural selection and information exchange between the populations, Civicioglu [26] proposed a novel population-based evolutionary algorithm called a backtracking search algorithm (BSA), which is a global searching technique to settle complex numerical optimization problems. In BSA, besides the famous operators used in the genetic algorithms (GAs) (i.e., the selection, mutation, and crossover operators), several particular mechanisms have also been

employed, such as a memory system in which a population generated from a randomly selected historical generation is stored. Specifically, there are two populations in the BSA. One is the historical population and the other is the evolution population. In each iteration, the historical population is updated through random selection from both the historical population and the evolution population. Then, a new temporary population, called the trial population, is generated based on the mutation and crossover mechanisms. Finally, the trial population is used to update the evolution population based on a greedy selection mechanism. According to the research conducted by Civicioglu [26], the implementation of BSA consists of five major processes: initialization, selection-I, mutation, crossover, and selection-II. These five stages are simply summarized as follows:

#### (a) Initialization

In this phase, individuals of the historical population  $oldPop$  and evolution population  $Pop$  are randomly initialized within the predefined search space using a uniform distribution  $U$  as follows:

$$\begin{aligned} Pop_{i,j} &= U\left(low_j, up_j\right), \quad i = 1, 2, \dots, N_{pop}; j = 1, 2, \dots, D \\ oldPop_{i,j} &= U\left(low_j, up_j\right), \end{aligned} \quad (6)$$

where  $N_{pop}$  and  $D$  are the size of population and the dimension, respectively; and  $[low_j, up_j]$  are the preset upper and lower boundaries of the variables to be optimized.

#### (b) Selection-I

In this stage, an option is provided to update the  $oldPop$  at the start of each iteration according to the following “if-then” rule:

$$\text{if } R_1 < R_2 \quad \text{then} \quad oldPop_{i,j} = Pop_{i,j}, \quad R_1, R_2 \in U(0, 1) \quad (7)$$

where  $R_1$  and  $R_2$  are two random numbers distributed uniformly from 0 to 1 to judge whether the historical population should be replaced by the evolution population in the current generation.

When  $oldPop$  is determined, the sequence of the individuals in  $oldPop$  is then changed by a random shuffling function  $permuting(\cdot)$ :

$$oldPop := permuting(oldPop) \quad (8)$$

where “ $:=$ ” indicates the update operator.

#### (c) Mutation

In this step, the temporary population, called trial population  $trialPop$ , is initialized using

$$\begin{aligned} trialPop &= Pop + F \cdot (oldPop - Pop) \\ F &= 3 \cdot rndn, \quad rndn \sim N(0, 1) \end{aligned} \quad (9)$$

where  $(oldPop - Pop)$  denotes the search direction matrix whose amplitude can be controlled by a control parameter  $F$ .

Due to the utilization of  $oldPop$  in the mutation operation, BSA can learn partial experiences from previous generations.

#### (d) Crossover

The final form of the trial population is determined in this stage. The crossover operator starts with a generation of a binary integer-valued matrix ( $map_{N_{pop} \times D}$ ) to determine which elements of population have to be manipulated. The crossover operator is realized using

$$trialPop_{ij} = \begin{cases} Pop_{ij}, & \text{if } map_{ij} = 1 \\ trialPop_{ij}, & \text{otherwise} \end{cases} \quad (10)$$

#### (e) Selection-II

In this phase, the population of the next generation is generated according to a greedy selection strategy. The trial individuals with better fitness values are used to update the corresponding individuals in population  $Pop_{ij}$ :

$$Pop_{ij} = \begin{cases} trialPop_{ij}, & \text{if } fitness(trialPop_{ij}) < fitness(Pop_{ij}) \\ Pop_{ij}, & \text{otherwise} \end{cases} \quad (11)$$

#### 2.4. The Proposed ELM-BSA Model for Flood Forecasting

As discussed in the introduction, ELM can save the calculation time by randomly generating network parameters instead of ardously tuning them. Compared with the traditional SLFNs with GL algorithms, ELM not only has a faster training speed and better generalization capability but also avoids the predefining computational parameters including the learning rate and stopping criteria. These advantages of ELM make it more suitable for solving the complex non-linear optimization problem, i.e., flood forecasting. Unfortunately, the random generation of input weights and hidden layer thresholds in ELM may provide some non-optimal or unnecessary network parameters which may reduce the prediction reliability, increase uncertainty of forecasting results, and produce unacceptable results for practical applications. To settle this problem, we proposed an ELM-BSA model, in which the input weights and thresholds of hidden layer neurons were optimized using BSA in the training period.

The construction of the ELM-BSA for flood forecasting is set to  $m-h-1$  due to there being only one node in the output layer. The implementation of the proposed model is described as follows:

Step 1: Normalize the original time series into the range [0, 1] using Equation (12), and then partition the normalized series into two parts: training and testing datasets.

$$Q_i^{nor} = \frac{Q_i - Q_{\min}}{Q_{\max} - Q_{\min}} \quad (12)$$

where  $Q_i^{nor}$  and  $Q_i$  are the normalized and observed streamflow, respectively; and  $Q_{\min}$  and  $Q_{\max}$  represent the minimum and maximum values of the original data, respectively.

Step 2: Initialize the related parameters of the proposed ELM-BSA model, such as the population size  $N_{pop}$  and the maximum iteration  $K$ .

Step 3: Define the architecture of the ELM and its activation function of hidden neurons, which is set to the sigmoid function in this study.

Step 4: Set the initial iteration number  $k = 1$ , and then initialize the historical population  $oldPop$  and evolution population  $Pop$  according to Equation (6). Each individual contains all parameters of the hidden layer, hence the  $i$ th individual in the  $k$ th generation can be written as

$$para(i, k) = [\omega_{1,(i,k)}^T, \omega_{2,(i,k)}^T, \dots, \omega_{h,(i,k)}^T, b_{1,(i,k)}, b_{2,(i,k)}, \dots, b_{h,(i,k)}] \quad (13)$$

where  $\omega_{1,(i,k)}^T, \omega_{2,(i,k)}^T, \dots, \omega_{h,(i,k)}^T$  represent the weight vector that connect the input nodes with the hidden layer neurons; and  $b_{1,(i,k)}, b_{2,(i,k)}, \dots, b_{h,(i,k)}$  are the thresholds for the hidden layer neurons.

Step 5: Calculate the output weights and initialize fitness values of all individuals of the population  $Pop$  using Equations (14) and (15), respectively.

$$H_{(i,k)} = \begin{bmatrix} \hat{\beta}_{(i,k)} = H_{(i,k)}^\dagger T \\ g(\omega_{1,(i,k)} X_1 + b_{1,(i,k)}) g(\omega_{2,(i,k)} X_1 + b_2) \cdots g(\omega_{h,(i,k)} X_1 + b_{h,(i,k)}) \\ g(\omega_{1,(i,k)} X_2 + b_{1,(i,k)}) g(\omega_{2,(i,k)} X_2 + b_2) \cdots g(\omega_{h,(i,k)} X_2 + b_{h,(i,k)}) \\ \vdots \quad \vdots \quad \vdots \\ g(\omega_{1,(i,k)} X_N + b_{1,(i,k)}) g(\omega_{2,(i,k)} X_N + b_2) \cdots g(\omega_{h,(i,k)} X_N + b_{h,(i,k)}) \end{bmatrix}_{N \times h} \quad (14)$$

$$f[para(i,k)] = \sqrt{\frac{1}{N} \sum_{j=1}^N (t_j - y_j)^2} = \sqrt{\frac{1}{N} \sum_{j=1}^N (t_j - \sum_{i=1}^h \beta_i g(\omega_i X_j + b_i))^2} \quad (15)$$

where  $H_{(i,k)}^\dagger$  is Moore–Penrose generalized inverse of the hidden-layer output matrix  $H_{(i,k)}$  for the  $i$ th individual in the  $k$ th generation;  $y_j$  and  $t_j$  are the calculated and target output in the training stage, respectively; and  $N$  is the total number of the training samples.

Step 6: Generate the historical population  $OldPop$  using the selection-I operator and obtain the initial form of the trial population  $trialPop$  using mutation operator.

Step 7: Apply the mutation operator on both the historical population and the trial population  $trialPop$  to generate the final form of the trial population.

Step 8: Calculate the fitness values of all individuals at the current generation, and then update individuals of the next generation through selection-II strategy.

Step 9: Set  $k = k + 1$ . If the maximum iteration is reached, go to Step 10; otherwise, go to Step 6.

Step 10: Apply the well-tuned ELM model to the forecasting phase using the validated dataset. Note, the output values of the forecasting model should be de-normalized to the range of the target output dataset.

## 2.5. Performance Indexes

Several indexes including coefficient of correlation ( $r$ ), Nash–Sutcliffe coefficient of efficiency (NSE), root mean square error (RMSE), and mean absolute error (MAE) were employed to evaluate the performance of the proposed model. Equations for these indexes are given as follows.

$$r = \left( \frac{\sum_{i=1}^N (Q_{obs,i} - \bar{Q}_{obs})(Q_{fore,i} - \bar{Q}_{fore})}{\sqrt{\sum_{i=1}^N (Q_{obs,i} - \bar{Q}_{obs})^2} \sqrt{\sum_{i=1}^N (Q_{fore,i} - \bar{Q}_{fore})^2}} \right), \quad -1 < r < 1 \quad (16)$$

$$\text{NSE} = 1 - \left( \frac{\sum_{i=1}^N (Q_{obs,i} - Q_{fore,i})^2}{\sum_{i=1}^N (Q_{obs,i} - \bar{Q}_{obs})^2} \right), \quad \text{NSE} \leq 1 \quad (17)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Q_{obs,i} - Q_{fore,i})^2}, \quad \text{RMSE} > 0 \quad (18)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |Q_{obs,i} - Q_{fore,i}|, \quad \text{MAE} > 0 \quad (19)$$

where  $Q_{obs,i}$  and  $Q_{fore,i}$  are the  $i$ th observed and predicted values of runoff, respectively;  $\bar{Q}_{obs}$  and  $\bar{Q}_{fore}$  are the average values of the observed and forecasted runoff, respectively; and  $N$  is the length of the data set.

Moreover, the Chinese flood forecasting standard recommend the use of the qualified rate (QR) to evaluate the flood forecasting performances [29]. A predicted peak value is regarded as “qualified” when the relative absolute error (RAE) between the predicted and the measured streamflow value is within the given threshold value [30]. The QR can be calculated using

$$\text{QR} = \frac{\sum_{i=1}^N \text{num}_i}{N} \times 100\% \quad \text{where}$$

$$\text{num}_i = \begin{cases} 1, & \text{if } (\text{RAE}_i \leq \varepsilon) \\ 0, & \text{otherwise} \end{cases}, \quad \text{RAE}_i = \frac{|Q_{\text{obs},i} - Q_{\text{fore},i}|}{Q_{\text{obs},i}} \quad (20)$$

where  $\text{RAE}_i$  is the relative absolute error (RAE) of the  $i$ th datum;  $\text{num}_i$  is set to 1 when RAE is less than or equal to the predefined threshold value ( $\varepsilon$ ), which is regarded as qualified forecasting. The  $\varepsilon$  is set to 20% in accordance with the Chinese forecasting standard (GB/T 22482-2008) [31].

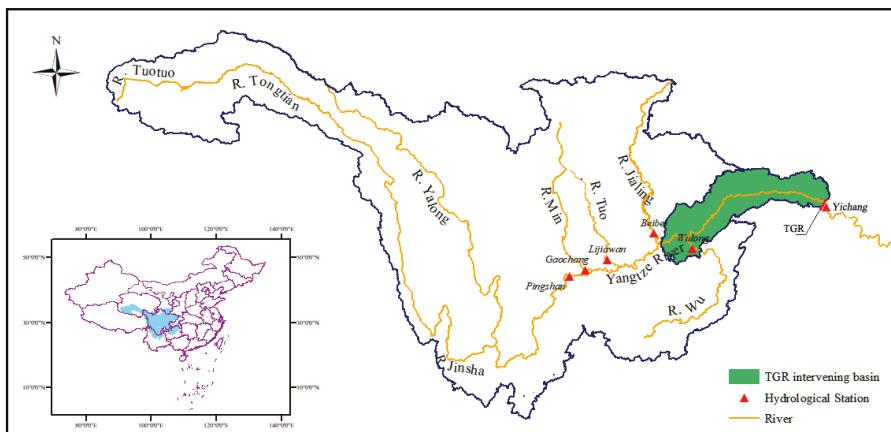
### 3. Case Study

#### 3.1. Study Area and Data

To validate the efficacy of the proposed model, the Yangtze River, which is the longest river in Asia and the third longest river in the world, was selected as a case study because abundant and detailed historical daily runoff data have been collected. The Yangtze River, which is nearly 6300 km long, originates from east of the Tibetan Plateau and flows eastward to the East China Sea in Shanghai city [10].

This study mainly focused on the upper Yangtze River, which covers a total area of nearly 1 million  $\text{km}^2$ , accounting for about 56% of the whole area of the Yangtze River, with a total length of 4529 km, up to 75% of the entire length of the Yangtze River. Flood events frequently occur in this region. During the historical years, extreme flood events, especially for the years 1870, 1954, 1998, 2010, and 2016, have caused heavy casualties and property losses. For example, in 2016, the whole Yangtze River basin suffered from a monstrous flood, which led to economic losses of 146.9 billion Chinese Yuan and affected nearly 60.74 million people [32,33]. Accordingly, flood forecasting is an essential task for modern flood prevention and disaster relief of the upper Yangtze River.

Floods in the Yangtze River usually occur in monsoon season between June and September. During this period, the temporal and spatial distribution characteristics of regional rainfall depend heavily on monsoon activities and seasonal movement of subtropical anticyclones. Floods in the middle-lower Yangtze River mainly come from the upper region of the Yichang Station, a control hydrological station of the Three Georges Reservoir (TGR) which is situated at an intersection point of the upstream Yangtze River and the middle reaches [34,35]. The main tributaries in the upper Yangtze River from upstream to downstream are Yalong, Min, Tuo, Jialing, and Wu Rivers as shown in Figure 1, where the control stations of each tributary are also given. In this study, the Jinsha River, rather than the Yalong River, was taken into account, because the Yalong River flows into the Jinsha River, which is considered part of the Yangtze River [10]. As shown in Figure 1, six gauging stations named Pingshan, Gaochang, Lijiawan, Beibei, Wulong, and Yichang located in these rivers were considered. Each of them has a concurrent mean daily flow data from the year 1998 to year 2007. The historical streamflow of Yichang Station and its upstream stations were taken as alternative input factors, and the streamflow of Yichang station at time  $t$  was considered as output. In other words, the proposed forecasting model aims to predict the outflow of the TGR. The data set was divided into subsets, in which the daily streamflow data from the year 1998 to 2005 was employed for model calibration, and the data from the year 2006 to 2007 for model validation.



**Figure 1.** Locations of hydrological stations in the study area.

### 3.2. Establishment of the Flood Forecasting Models

Determination of model inputs is the most significant step for the data-driven forecasting model. The data-driven approaches may provide unreliable results when the inputs contain irrelevant or redundant information. However, there is no uniform approach to determine the input variables. According to a review conducted by Bowden et al. [11], the major approaches for input determination/selection in hydrological forecasting can be divided into three groups: trial and error method, linear method, and non-linear method. Considering the demerits and merits of these methods, a linear method called partial cross-correlation (PCC) [11] and a nonlinear approach called entropy based-partial mutual information (PMI) proposed by Chen et al. [10] were selected and compared. In the entropy based-PMI method, entropy theory, a famous tool to derive distribution functions [36,37], was combined with copula functions to pre digest the solving process of PMI. Therefore, using these three techniques, seven different input combination schemes were obtained as shown in Table 1, where  $\varphi(\cdot)$  indicates the complicated nonlinear mapping function between the input factors and the output results and  $Q_{ps}$ ,  $Q_{gc}$ ,  $Q_{ljw}$ ,  $Q_{bb}$ ,  $Q_{wl}$ , and  $Q_{yc}$  indicate the streamflow of the Pingshan, Gaochang, Lijianwan, Beibei, Wulong, and Yichang gauging stations, respectively, and  $t$  represents the current time.

**Table 1.** Different input sets calculated by trial and error, PCC and PMI approaches.

Schemes	Number of Input Variables	Established Models
M1	1	$Q_{yc}(t) = \varphi[Q_{yc}(t-1)]$
M2	2	$Q_{yc}(t) = \varphi[Q_{yc}(t-1), Q_{yc}(t-2)]$
M3	3	$Q_{yc}(t) = \varphi[Q_{yc}(t-1), Q_{yc}(t-2), Q_{yc}(t-3)]$
M4	4	$Q_{yc}(t) = \varphi[Q_{yc}(t-1), Q_{yc}(t-2), Q_{yc}(t-3), Q_{yc}(t-4)]$
M5	5	$Q_{yc}(t) = \varphi[Q_{yc}(t-1), Q_{yc}(t-2), Q_{yc}(t-3), Q_{yc}(t-4), Q_{yc}(t-5)]$
M6	6	$Q_{yc}(t) = \varphi \left[ Q_{yc}(t-1), Q_{wl}(t-4), Q_{bb}(t-3), Q_{ljw}(t-2), Q_{gc}(t-2), Q_{ps}(t-1) \right]$
M7	7	$Q_{yc}(t) = \varphi \left( Q_{yc}(t-1), Q_{yc}(t-2), Q_{wl}(t-2), Q_{bb}(t-2), Q_{ljw}(t-3), Q_{gc}(t-3), Q_{ps}(t-1) \right)$

The input sets of the first five schemes M1 to M5 were designed according to the trial and error method, and schemes M6 and M7 were determined by Chen et al. [10] based on the PCC and PMI approaches, respectively. It can be seen that the first five schemes M1 to M5 only considered the historical runoff of the Yichang station ( $Q_{yc}$ ), whereas schemes M6 and M7 used both the anterior runoff from the Yichang station and those from all control stations of the main tributaries located on

the upper Yangtze River as input variables. All of the seven input sets were fed into ELM-BSA, GRNN, and ELM models to train.

In addition, the number of hidden neurons also plays an important role for establishment of the forecasting models. To obtain the suitable number of hidden neurons, a grid search algorithm was employed in this study. For the proposed ELM-BSA model, the parameters of the BSA were set to  $N_{pop} = 30$  and  $K = 100$ . All forecasting models established in this study were encoded based on the Matrix Laboratory (MATLAB R2015a) platform manufactured by Mathwork Incoperation, Springfield, MA, USA.

### 3.3. Sensitivity Analysis of Different Input Sets

To testify the efficiency of the proposed ELM-BSA model, the GRNN and ELM models were selected as benchmark models. Input selection is one of the important steps for flood forecasting based on the data-driven method. Hence, all seven input schemes mentioned in Table 1 were taken into account in this study. The GRNN, ELM, and the proposed ELM-BSA models were employed for flood forecasting of the Yichang station located on the Yangtze River. Five performance indexes were used to evaluate the efficiency of the above three forecasting models. The data set was divided into two sub-sets. The first 8 years (from the year 1998 to year 2005) was used for model calibration and the remaining 2 years (from the year 2006 to year 2007) were used for model validation. Results of the three models for both the training and testing periods are given in Table 2, where the model with the best performance is highlighted in bold. It can be seen that compared with the GRNN and ELM models, the proposed ELM-BSA model performed better based on the values of the three indexes, no matter what the input combinations were. The most appropriate model inputs were not the same for the three forecasting models and the response of each forecasting model was not identical when using the same input sets. In other words, accurate forecasting results were not only affected by the inputs, but also by the model structure and its corresponding parameters. This also indicates that obtaining the accurate flood forecasting results is a complicated and challenging task under the comprehensive effects of model inputs, structures, and parameters.

**Table 2.** Performances of the ELM-BSA, ELM, and GRNN models in both the training and testing periods.

Schemes	Training Period					Testing Period				
	<i>r</i>	NSE	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	QR	<i>r</i>	NSE	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	QR
<b>GRNN</b>										
M1	0.9684	0.9377	2734	1939	0.9632	0.9598	0.9183	2735	1865	0.8319
M2	<b>0.9791</b>	<b>0.9584</b>	<b>2234</b>	<b>1573</b>	<b>0.9800</b>	<b>0.9645</b>	<b>0.9271</b>	<b>2583</b>	<b>1792</b>	<b>0.8571</b>
M3	0.9264	0.8579	4128	3012	0.8319	0.8925	0.7759	4530	3159	0.6597
M4	0.8605	0.7399	5585	4152	0.6828	0.8047	0.6062	6006	4374	0.5084
M5	0.7950	0.6314	6649	4972	0.6166	0.7195	0.4844	6872	5098	0.3992
M6	0.9793	0.9589	2220	1592	0.9664	0.9642	0.9191	2722	1907	0.8319
M7	0.9781	0.9565	2283	1617	0.9737	0.9562	0.9111	2853	1879	0.8487
<b>ELM</b>										
M1	0.9681	0.9371	2746	1956	0.9674	0.9611	0.9226	2663	1715	0.9286
M2	<b>0.9778</b>	<b>0.9561</b>	<b>2294</b>	<b>1562</b>	<b>0.9706</b>	<b>0.9729</b>	<b>0.9440</b>	<b>2265</b>	<b>1457</b>	<b>0.9580</b>
M3	0.9187	0.8439	4327	3088	0.8508	0.9010	0.8019	4260	2823	0.7311
M4	0.8471	0.7175	5821	4271	0.6859	0.8123	0.6347	5785	4036	0.5420
M5	0.7807	0.6094	6845	5139	0.5809	0.7298	0.4916	6824	4983	0.4370
M6	0.9742	0.9490	2473	1788	0.9674	0.9681	0.9320	2495	1747	0.9076
M7	0.9771	0.9547	2331	1608	0.9664	0.9724	0.9415	2315	1538	0.9160

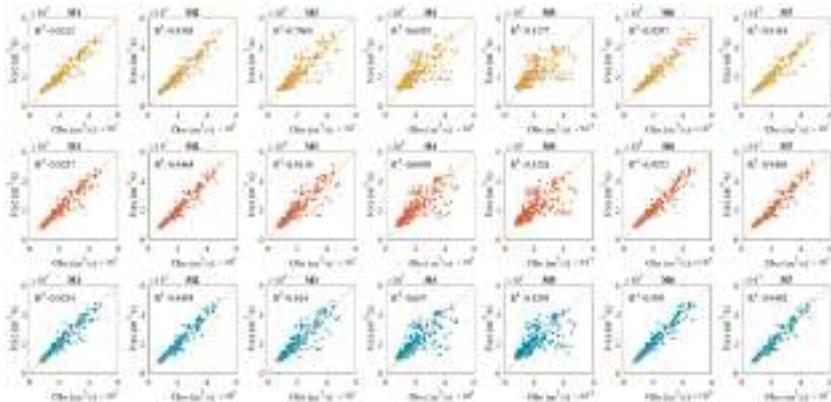
**Table 2.** Cont.

Schemes	Training Period					Testing Period				
	r	NSE	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	QR	r	NSE	RMSE (m <sup>3</sup> /s)	MAE (m <sup>3</sup> /s)	QR
<b>ELM-BSA</b>										
M1	0.9681	0.9372	2745	1957	0.9622	0.9609	0.9222	2669	1729	0.9286
M2	0.9787	0.9578	2251	1519	0.9685	0.9743	0.9477	2188	1390	0.9454
M3	0.9199	0.8461	4296	3062	0.8424	0.9022	0.8046	4231	2804	0.7311
M4	0.8497	0.7220	5775	4227	0.6933	0.8106	0.6328	5800	3978	0.5798
M5	0.7853	0.6167	6780	5093	0.5945	0.7276	0.4907	6830	4900	0.4580
M6	0.9747	0.9501	2447	1762	0.9643	0.9690	0.9340	2458	1627	0.9328
M7	<b>0.9787</b>	<b>0.9578</b>	<b>2250</b>	<b>1516</b>	<b>0.9706</b>	<b>0.9743</b>	<b>0.9477</b>	<b>2188</b>	<b>1388</b>	<b>0.9454</b>

It can be seen from Table 2 that when the GRNN model was used, the model with the M2 input set produced the best forecasting results in both the training and validation periods. Similarly, the ELM based on the M2 yielded the best forecasting results for both the training and testing periods. For the proposed ELM-BSA method, it demonstrated that the model with the M7 input sets showed better performances. Overall, the most suitable input sets for the GRNN, ELM, and ELM-BSA models were M2, M2, and M7 respectively.

To further compare the predicted streamflow with the observed flow, the predicted and observed flow were drawn in the same figure as shown in Figure 2, where the x-axis represents the observed flow and the y-axis represents the predicted flow. If the model works well, the predicted flow should be equal to the observed flow. Results of the three flood forecasting models with seven input schemes M1–M7 in the validation period are shown in Figure 2. The regression coefficient  $R^2$  was also calculated and displayed in Figure 2. If the predicted and observed streamflow being compared are similar, the scatter points should approximately lie on the line  $y = x$ , namely the diagonal line shown in Figure 2. It can be seen that according to the  $R^2$  and fitting results, the input schemes M1, M2, M6, and M7 for both of the three models could always provide better results than other input schemes. For the forecasting models based on the input set selected by the PMI method, M7 provided slightly better results than those based on the inputs chosen by the PCC approach, namely M6. It can also be seen from Table 2 and Figure 2 that the three models with input schemes M1 and M2 showed better performances than those models with the schemes M3 to M5. This means that when more anterior flows, such as the flows at lag time t-3, t-4, and t-5, are considered, the performance of the models became worse, which means more inputs bring noise to the forecasting system. Meanwhile, models based on different input sets yielded different results, while the best input sets were not identical for all forecasting models. According to the results of Figure 2 and Table 2, the best input combinations for the GRNN, ELM, and ELM-BSA models were M2, M2, and M7, respectively. Figure 2 also demonstrates that the proposed ELM-BSA model with the M7 input set performed best among all the combinations of inputs and models with the  $R^2$  value of 0.9492.

Table 3 summarizes the best performance results calculated using the three models with different input sets. It indicates that compared with other methods, there were significant improvements when the ELM-BSA was used. The ELM-BSA model provided better forecasting results than the GRNN and ELM models for daily streamflow forecasting. For the validation period, compared with the GRNN model, when the ELM-BSA model was used, the performance indexes  $r$ , NSE, RMSE, and QR increased by 1.05%, 3.12%, and 13.64%, respectively, and the indexes RMSE and MAE decreased by 19.63% and 27.22%, respectively. Similarly, compared with the standard ELM model, when the ELM-BSA was used, the indexes  $r$ , NSE, RMSE, and QR increased by 0.15%, 0.4%, and 1.32%, respectively, and the indexes RMSE and MAE decreased by 3.42% and 4.72%, respectively. Therefore, the proposed method increased the flood forecasting model accuracy.



**Figure 2.** Scatter plots of observed (Obs) and predicted (Fore) runoff provided by the GRNN (the first row), ELM (the second row), and ELM-BSA (the third row) models with different input sets.

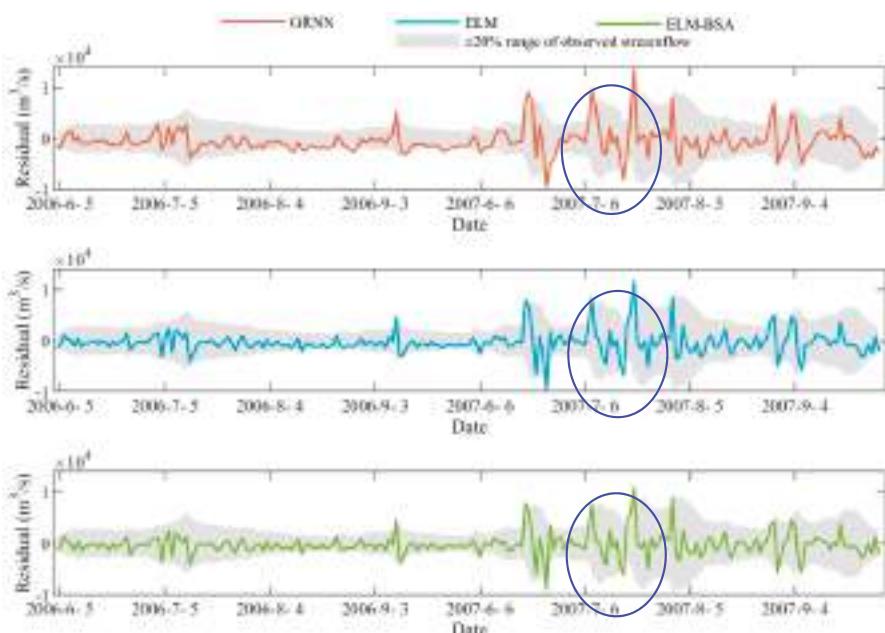
**Table 3.** The performance of the best GRNN, ELM, and ELM-BSA models for flood forecasting at the Yichang station.

Model	R	NSE	RMSE ( $\text{m}^3/\text{s}$ )	MAE ( $\text{m}^3/\text{s}$ )	QR
GRNN (M6)	0.9642	0.9191	2722	1906.7	0.8319
ELM (M2)	0.9729	0.9440	2265	1456.5	0.9580
ELM-BSA (M7)	0.9743	0.9477	2188	1387.7	0.9454
Improvement (ELM-BSA vs. GRNN, %)	1.05	3.12	19.63	27.22	13.64
Improvement (ELM-BSA vs. ELM, %)	0.15	0.40	3.42	4.72	1.32

As streamflow in the flood season has a great impact on the scientific decision-making of modern water resources management and planning, the number of forecasting values whose relative error beyond the specific range ( $\pm 15\%$ ,  $\pm 20\%$ , and  $\pm 25\%$ ) are given in Table 4, where the number and proportion of over-ranging points for each forecasting model in the testing period are shown. Results indicate that the total number of over-ranging points of the ELM-BSA model was always less than the other two models for each specific range. This means that the ELM-BSA model performed better than GRNN and ELM for the daily streamflow forecasting. The advantages of the ELM-BSA model for high streamflow forecasting can be visually seen in Figure 3, where the residual values of the best ELM-BSA, GRNN, and ELM models in the validation period are presented, and the  $\pm 20\%$  intervals of the observed streamflow is also presented. Results show that the ELM-BSA produced the best performance because it provided fewer residual values falling outside the  $\pm 20\%$  range than the other two models. For example, its residual value out of the reference range between the date 6 July 2007, and 5 August 2007 (marked in Figure 3) was comparatively less serious. Meanwhile, the ELM-BSA model produced smaller maximum residual values than the other two models, while the GRNN performed even worse than the ELM. Additionally, the GRNN model was not suitable for the low and high streamflow parts due to its remarkable over-estimation and under-estimation. All these results imply that the proposed model was superior to the other models for flood forecasting.

**Table 4.** Number of forecasting values whose relative error was beyond the specific range.

Model	GRNN		ELM		ELM-BSA	
	Number	Proportion	Number	Proportion	Number	Proportion
Beyond $\pm 15\%$	59	24.79	54	22.69	26	10.92
Beyond $\pm 20\%$	34	14.29	22	9.24	13	5.46
Beyond $\pm 25\%$	20	8.40	8	3.36	8	3.36

**Figure 3.** Residual values of the three models in the validation period.

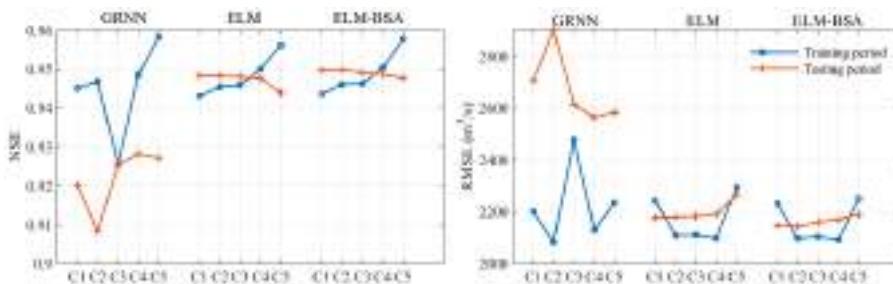
### 3.4. Sensitivity Analysis of Different Training Sample Sizes

Another important factor affecting the forecasting accuracy of data-driven forecasting models is the number of training samples. Hence, in this sub-section, five schemes were designed and employed to further test the performances of the proposed ELM-BSA model with different training data sizes. In each case, the same dataset, the data from the last two years (from the year 2006 to year 2007), was used for model validation. Performances of the ELM-BSA model in these five scenarios are given in Table 5. Meanwhile, Figure 4 shows the values of indexes RMSE and NSE calculated using ELM-BSA with different training data sizes. ELM-BSA with a different number of training data demonstrate different forecasting results and all these results can comply with the Chinese flood forecasting standard [31]. Hence, these models developed in this study can be applied to practical use. Meanwhile, the forecasting accuracies of the ELM-BSA model were always better than the other two models in all cases, because the ELM-BSA model could yield the largest NSE values and the lowest RMSE values in the validation period among these three forecasting models. In the training period, the forecasting accuracies grew with the increase of training data size, except for the GRNN model with Case 3. In the validation period, the ELM and ELM-BSA models provided stable NSE values for Cases 1–4, while there is a sudden drop of NSE in Case 5 for ELM. The forecasting accuracies of GRNN in the validation period increased with the increase of the training samples, except for Case 2. The ELM and ELM-BSA could generate stable RMSE values for Cases 2–4 in the training period and for

Cases 1–4 in the testing period. As for GRNN, its performances in the testing stage seemed to be better when the training samples were increased, whereas its performance fluctuated in the training period with an increase of the training samples. Additionally, Figure 4 clearly shows that the training number in the Case 3 was the best one for all the forecasting models, because in this condition, the accuracies in both training and testing periods for every forecasting model were well-balanced. These results indicate that more samples adopted to train forecasting model may be conducive to enhancing the forecasting accuracy of the training stage but may be detrimental to the performance in the testing phase in the condition where the number of training samples exceeds a specific range. Therefore, in the real engineering applications, it is important to balance the sample sizes of the training and testing datasets, which will be helpful to promote the robustness and accuracy of the forecasting models. All the above results prove the superiority of the ELM-BSA model in the aspects of both robustness and accuracy when compared with the other two widely used forecasting models. This is due to the fact that the ELM-BSA model processes the merits of both the BSA and ELM, which enhances its generalization ability and robustness.

**Table 5.** Results of GRNN, ELM, and ELM-BSA in five cases with different training sample sizes.

Case	Year	Period	r	NSE	RMSE ( $\text{m}^3/\text{s}$ )	MAE ( $\text{m}^3/\text{s}$ )	QR
<b>GRNN</b>							
Case 1	1994–2005	training	0.9723	0.9451	2200	1507	0.9664
		testing	0.9611	0.9201	2705	1924	0.8151
Case 2	1995–2005	training	0.9731	0.9469	2082	1474	0.9681
		testing	0.9561	0.9082	2900	1970	0.8193
Case 3	1996–2005	training	<b>0.9650</b>	<b>0.9255</b>	<b>2477</b>	<b>1742</b>	<b>0.9636</b>
		testing	<b>0.9658</b>	<b>0.9255</b>	<b>2613</b>	<b>1911</b>	<b>0.8109</b>
Case 4	1997–2005	training	0.9740	0.9485	2130	1499	0.9856
		testing	0.9652	0.9282	2565	1806	0.8277
Case 5	1998–2005	training	0.9791	0.9584	2234	1573	0.9800
		testing	0.9645	0.9271	2583	1792	0.8571
<b>ELM</b>							
Case 1	1994–2005	training	0.9711	0.9431	2241	1479	0.9517
		testing	0.9741	0.9483	2175	1367	0.9538
Case 2	1995–2005	training	0.9724	0.9455	2108	1414	0.9580
		testing	0.9741	0.9482	2178	1385	0.9538
Case 3	1996–2005	training	<b>0.9726</b>	<b>0.9459</b>	<b>2110</b>	<b>1402</b>	<b>0.9608</b>
		testing	<b>0.9740</b>	<b>0.9481</b>	<b>2181</b>	<b>1379</b>	<b>0.9538</b>
Case 4	1997–2005	training	0.9747	0.9500	2097	1409	0.9664
		testing	0.9739	0.9477	2189	1380	0.9580
Case 5	1998–2005	training	0.9778	0.9561	2294	1562	0.9706
		testing	0.9729	0.9440	2265	1457	0.9580
<b>ELM-BSA</b>							
Case 1	1994–2005	training	0.9714	0.9436	2231	1465	0.9517
		testing	0.9747	0.9496	2148	1352	0.9454
Case 2	1995–2005	training	0.9727	0.9461	2097	1391	0.9597
		testing	0.9748	0.9498	2143	1351	0.9454
Case 3	1996–2005	training	<b>0.9728</b>	<b>0.9463</b>	<b>2103</b>	<b>1392</b>	<b>0.9594</b>
		testing	<b>0.9745</b>	<b>0.9491</b>	<b>2159</b>	<b>1368</b>	<b>0.9454</b>
Case 4	1997–2005	training	0.9748	0.9503	2091	1405	0.9652
		testing	0.9745	0.9486	2169	1380	0.9454
Case 5	1998–2005	training	0.9787	0.9578	2250	1516	0.9706
		testing	0.9743	0.9477	2188	1388	0.9454



**Figure 4.** NSE and RMSE values of ELM-BSA, GRNN, and ELM in five cases with different training sample sizes.

In summary, all the above results obtained from Sections 3.3 and 3.4 indicate that the ELM-BSA model is a powerful tool to model the daily streamflow and can produce more reliable performance compared with GRNN and ELM. It provides an effective alternative for flood forecasting.

#### 4. Conclusions

Reliable and robust flood forecasting plays an essential role in effective/scientific flood control and many activities associated with water resources management. On the basis of an extreme learning machine (ELM) and an emerging dual population-based evolutionary algorithm named backtracking search optimization algorithm (BSA), this paper developed an improved extreme learning machine named ELM-BSA for short-term flood forecasting. In the new forecasting model, BSA was used to find the appropriate hidden node parameters of ELM, and then the well-tuned ELM was applied to do one-step-ahead forecasting. For the purpose of evaluating the performance of the developed model, the standard ELM and a widely-used GRNN model were taken as reference models. The upper Yangtze River was selected as a case study. Experiments with different input combination schemes and training sample sizes indicated that the proposed ELM-BSA model was superior to the current ELM and GRNN models. For example, compared with the GRNN model, the improvements achieved by the ELM-BSA model regarding the indexes NSE and RMSE values in the validation period were 3.12% and 19.63%, respectively. Moreover, when the sample size changed within a proper range, the accuracy of the developed model fluctuated in a smaller scope than those of the ELM and GRNN models, which demonstrated the stability of the proposed model. Therefore, the ELM-BSA model is a powerful tool for flood forecasting. It is necessary to apply this new method to real-time flood forecasting.

**Author Contributions:** L.C. designed this experiment and write parts of the paper; N.S. and Q.Z. did the calculation work and wrote parts of the paper. C.Z., J.Z. (Jianzhong Zhou), Y.Z., J.Z. (Junhong Zhang) made some corrections to the paper.

**Funding:** This research was funded by the National Key R & D Program of China (2017YFC0405900) and the National Natural Science Foundation of China (91547208; 51879109).

**Acknowledgments:** The authors also greatly appreciate the anonymous reviewers and academic editor for their careful comments and valuable suggestions to improve the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

- Yaseen, Z.M.; El-Shafie, A.; Jaafar, O.; Afan, H.A.; Sayl, K.N. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *J. Hydrol.* **2015**, *530*, 829–844. [[CrossRef](#)]
- Chen, L.; Singh, V.P.; Guo, S. Measure of correlation between river flows using the copula-entropy method. *J. Hydrol. Eng.* **2013**, *18*, 1591–1606. [[CrossRef](#)]
- Chen, L.; Singh, V.P.; Lu, W.; Zhang, J.; Zhou, J.; Guo, S. Streamflow forecast uncertainty evolution and its effect on real-time reservoir operation. *J. Hydrol.* **2016**, *540*, 712–726. [[CrossRef](#)]

4. Huang, K.; Ye, L.; Chen, L.; Wang, Q.; Dai, L.; Zhou, J.; Singh, V.P.; Huang, M.; Zhang, J. Risk analysis of flood control reservoir operation considering multiple uncertainties. *J. Hydrol.* **2018**, *565*, 672–684. [[CrossRef](#)]
5. Costabile, P.; Costanzo, C.; Macchione, F. A storm event watershed model for surface runoff based on 2D fully dynamic wave equations. *Hydrol. Process.* **2013**, *27*, 554–569. [[CrossRef](#)]
6. Rousseau, M.; Cerdan, O.; Delestre, O.; Dupros, F.; James, F.; Cordier, S. Overland flow modeling with the shallow water equations using a well-balanced numerical scheme: Better predictions or just more complexity. *J. Hydrol. Eng.* **2015**, *20*, 04015012. [[CrossRef](#)]
7. Bout, B.; Jetten, V.G. The validity of flow approximations when simulating catchment-integrated flash floods. *J. Hydrol.* **2018**, *556*, 674–688. [[CrossRef](#)]
8. Bellos, V.; Tsakiris, G. A hybrid method for flood simulation in small catchments combining hydrodynamic and hydrological techniques. *J. Hydrol.* **2016**, *540*, 331–339. [[CrossRef](#)]
9. Zhou, J.; Peng, T.; Zhang, C.; Sun, N. Data pre-analysis and ensemble of various artificial neural networks for monthly streamflow forecasting. *Water* **2018**, *10*, 628. [[CrossRef](#)]
10. Chen, L.; Ye, L.; Singh, V.; Asce, F.; Zhou, J.; Guo, S. Determination of input for artificial neural networks for flood forecasting using the copula entropy method. *J. Hydrol. Eng.* **2014**, *19*, 217–226. [[CrossRef](#)]
11. Bowden, G.J.; Dandy, G.C.; Maier, H.R. Input determination for neural network models in water resources applications. Part 1—Background and methodology. *J. Hydrol.* **2005**, *301*, 75–92. [[CrossRef](#)]
12. Chang, F.-J.; Tsai, M.-J. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *J. Hydrol.* **2016**, *535*, 256–269. [[CrossRef](#)]
13. Chang, F.J.; Lai, H.C. Adaptive neuro-fuzzy inference system for the prediction of monthly shoreline changes in northeastern taiwan. *Ocean Eng.* **2014**, *84*, 145–156. [[CrossRef](#)]
14. Zhou, Y.; Chang, F.J.; Guo, S.; Ba, H.; He, S. A robust recurrent anfis for modeling multi-step-ahead flood forecast of three gorges reservoir in the yangtze river. *Hydrol. Earth Syst. Sci. Discuss.* **2017**, *1*–29. [[CrossRef](#)]
15. Xing, B.; Gan, R.; Liu, G.; Liu, Z.; Zhang, J.; Ren, Y. Monthly mean streamflow prediction based on bat algorithm-support vector machine. *J. Hydrol. Eng.* **2016**, *21*, 04015057. [[CrossRef](#)]
16. Tayyab, M.; Zhou, J.; Dong, X.; Ahmad, I.; Sun, N. Rainfall-runoff modeling at jinsha river basin by integrated neural network with discrete wavelet transform. *Meteorol. Atmos. Phys.* **2017**, *129*, 1–11. [[CrossRef](#)]
17. Peng, T.; Zhou, J.; Zhang, C.; Fu, W. Streamflow forecasting using empirical wavelet transform and artificial neural networks. *Water* **2017**, *9*, 406. [[CrossRef](#)]
18. Cheng, C.; Niu, W.; Feng, Z.; Shen, J.; Chau, K. Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization. *Water* **2015**, *7*, 4232–4246. [[CrossRef](#)]
19. Chang, F.J.; Chen, P.A.; Lu, Y.R.; Huang, E.; Chang, K.Y. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* **2014**, *517*, 836–846. [[CrossRef](#)]
20. Deo, R.C.; Sahin, M. An extreme learning machine model for the simulation of monthly mean streamflow water level in eastern queensland. *Environ. Monit. Assess.* **2016**, *188*, 90. [[CrossRef](#)] [[PubMed](#)]
21. Zhou, J.; Sun, N.; Jia, B.; Tian, P. A novel decomposition-optimization model for short-term wind speed forecasting. *Energies* **2018**, *11*, 1572. [[CrossRef](#)]
22. Li, C.; Xiao, Z.; Xia, X.; Zou, W.; Zhang, C. A hybrid model based on synchronous optimisation for multi-step short-term wind speed forecasting. *Appl. Energy* **2018**, *215*, 131–144. [[CrossRef](#)]
23. Huang, G.; Zhu, Q.; Siew, C. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
24. Yaseen, Z.M.; Jaafar, O.; Deo, R.C.; Kisi, O.; Adamowski, J.; Quilty, J.; El-Shafie, A. Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in iraq. *J. Hydrol.* **2016**, *542*, 603–614. [[CrossRef](#)]
25. Han, F.; Yao, H.F.; Ling, Q.H. An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing* **2013**, *116*, 87–93. [[CrossRef](#)]
26. Civicioglu, P. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8144. [[CrossRef](#)]
27. Chen, L.; Singh, V.P.; Guo, S.; Zhou, J.; Ye, L. Copula entropy coupled with artificial neural network for rainfall-runoff simulation. *Stoch. Environ. Res. Risk Assess.* **2014**, *28*, 1755–1767. [[CrossRef](#)]
28. Hosseini, S.M.; Mahjouri, N. Integrating support vector regression and a geomorphologic artificial neural network for daily rainfall-runoff modeling. *Appl. Soft Comput.* **2016**, *38*, 329–345. [[CrossRef](#)]

29. Chen, L.; Zhang, Y.; Zhou, J.; Singh, V.P.; Guo, S.; Zhang, J. Real-time error correction method combined with combination flood forecasting technique for improving the accuracy of flood forecasting. *J. Hydrol.* **2014**, *521*, 157–169. [[CrossRef](#)]
30. Li, X.; Guo, S.; Liu, P.; Chen, G. Dynamic control of flood limited water level for reservoir operation by considering inflow uncertainty. *J. Hydrol.* **2010**, *391*, 126–134. [[CrossRef](#)]
31. Ministry of Water Resources (MWR). *Regulation for Calculating Design Flood of Water Resources and Hydropower Projects*; Chinese Shuili Shuidian Press: Beijing, China, 2008. (In Chinese)
32. Huang, K.D.; Chen, L.; Zhou, J.; Zhang, J.; Singh, V.P. Flood hydrograph coincidence analysis for mainstream and its tributaries. *J. Hydrol.* **2018**, *565*, 341–353. [[CrossRef](#)]
33. Zhou, C.; Sun, N.; Chen, L.; Ding, Y.; Zhou, J.; Zha, G.; Luo, G.; Dai, L.; Yang, X. Optimal operation of cascade reservoirs for flood control of multiple areas downstream: A case study in the upper Yangtze river basin. *Water* **2018**, *10*, 1250. [[CrossRef](#)]
34. Chen, L.; Singh, V.P.; Guo, S.L.; Hao, Z.C.; Li, T.Y. Flood coincidence risk analysis using multivariate copula functions. *J. Hydrol. Eng.* **2012**, *17*, 742–755. [[CrossRef](#)]
35. Chen, L.; Singh, V.P.; Guo, S.; Zhou, J.; Zhang, J. Copula-based method for multisite monthly and daily streamflow simulation. *J. Hydrol.* **2015**, *528*, 369–384. [[CrossRef](#)]
36. Chen, L.; Singh, V.; Huang, K. Bayesian technique for the selection of probability distributions for frequency analyses of hydrometeorological extremes. *Entropy* **2018**, *20*, 117. [[CrossRef](#)]
37. Chen, L.; Singh, V.P. Entropy-based derivation of generalized distributions for hydrometeorological frequency analysis. *J. Hydrol.* **2018**, *557*, 699–712. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Dongting Lake Water Level Forecast and Its Relationship with the Three Gorges Dam Based on a Long Short-Term Memory Network

Chen Liang <sup>1</sup>, Hongqing Li <sup>2</sup>, Mingjun Lei <sup>3</sup> and Qingyun Du <sup>1,4,5,\*</sup>

<sup>1</sup> School of Resource and Environmental Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; chenliang90210@whu.edu.cn

<sup>2</sup> Changjiang Water Resources Protection Institute, Wuhan 430010, China; hqli88@163.com

<sup>3</sup> Yangtze River Water Resources Protection Bureau, Wuhan 430010, China; mjlei@163.com

<sup>4</sup> Key Laboratory of Geographic Information System, Ministry of Education, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

<sup>5</sup> Key Laboratory of Digital Mapping and Land Information Application Engineering, National Administration of Surveying, Mapping and Geo-information, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

\* Correspondence: qydu@whu.edu.cn; Tel.: +86-27-6877-8842

Received: 17 August 2018; Accepted: 1 October 2018; Published: 4 October 2018

**Abstract:** To study the Dongting Lake water level variation and its relationship with the upstream Three Gorges Dam (TGD), a deep learning method based on a Long Short-Term Memory (LSTM) network is used to establish a model that predicts the daily water levels of Dongting Lake. Seven factors are used as the input for the LSTM model and eight years of daily data (from 2003 to 2012) are used to train the model. Then, the model is applied to the test dataset (from 2011 to 2013) for forecasting and is evaluated using the root mean squared error (RMSE) and the coefficient of determination ( $R^2$ ). The test shows the LSTM model has better accuracy compared to the support vector machine (SVM) model. Furthermore, the model is adjusted to simulate the situation where the TGD does not exist to explore the dam's impact. The experiment shows that the water level of Dongting Lake drops conspicuously every year from September to November during the TGD impounding period, and the water level increases mildly during dry seasons due to TGD replenishment. Additionally, the impact of the TGD results in a water level decline in Dongting Lake during flood peaks and a subsequent lagged rise. This research provides a tool for flood forecasting and offers a reference for TGD water regulation.

**Keywords:** deep learning; LSTM network; water level forecast; the Three Gorges Dam; Dongting Lake

## 1. Introduction

The large freshwater lakes of the world are an extremely valuable resource, not only because 68% of the global liquid surface freshwater is contained within lakes but also because of their importance to the economies, social structure, and viability of riparian countries [1]. Dongting Lake is the second largest freshwater lake in China and is also renowned for its wetland resources [2]. Wetlands are an area of transition between dry land and water bodies, and wetlands are often described as kidneys of the earth for their great contributions to flood control, groundwater replenishment, water purification, agriculture, and biological diversity [3]. Wetlands vary seasonally because water bodies change dramatically between dry and wet seasons. Therefore, the change in water levels could influence the biodiversity community patterns and functions in lake ecosystems [4]. Thus, it is of great significance to study lake water level variations. In this case, when studying the lake water level changes in Dongting Lake, consideration of the influence of the upstream Three Gorges Dam (TGD) is inevitable.

The TGD is a hydroelectric gravity dam that spans the Yangtze River in the town of Sandouping, Yiling District, Yichang, Hubei Province, China. The TGD is the world's largest water conservancy project with a total reservoir storage capacity of 39.3 billion m<sup>3</sup> [5]. Like any other hydraulic project in the world, the TGD has a large impact on the surrounding geological and environmental systems. Its function of flood control and hydroelectric generation can alter the downstream hydrologic regime by affecting the streamflow of the Yangtze River, the total runoff quantity, water quality, and duration of extreme runoff [6,7]. Additionally, as one of the most controversial projects in the world, the TGD has impacted the landscape, wildlife, agriculture, and other areas [8]. For Dongting Lake, the river-lake relationship becomes increasingly complicated [9], which makes it increasingly challenging to determine how the water level of Dongting Lake is affected by the TGD; however, this relationship is worthy of study.

The research on lake water level variations has a long history, and a considerable number of cases have been studied. The methods used in these studies can be summarized into two categories: physics-based methods and data-driven methods [10–12]. Physics-based methods analyze the lake water level based on the physical process, which is often completed through solving hydrodynamic equations. For example, Lai et al. [13] applied the coupled hydrodynamic analysis model to the middle Yangtze River to compute the variation in the water regime induced by water storage. Wu et al. [14] conducted physical model experiments to study the effects of the TGD on the water level in Lake Poyang, which is in the lower reach of the Yangtze River. Jiang and Huang [15] used the Saint Venant equations of river dynamics to study the impacts of the TGD project on the water level of Chenglingji station through modeling of the hydrologic process of the Yangtze River and used this model to predict the water level at Chenglingji station. Data-driven methods involve the use of scientific computing models to simulate the relationship between lake water level and its influencing factors. Different kinds of models have been constructed to stimulate certain scenarios. For example, Liu et al. [16] proposed a multivariate conditional model based on copulas for streamflow prediction and the refinement of spatial precipitation estimates and compared the model with support vector regression (SVR) and the adaptive neuro-fuzzy inference system (ANFIS). Khedun et al. [17] used a copula-based model to examine the dependence structure between the large-scale climate indices and average monthly seasonal precipitation and then used it to forecast precipitation anomalies in different climate divisions of Texas, USA. Liu et al. [18] developed a Bayesian wavelet-support vector regression model (BWS model) using local meteohydrological observations and climate indices as potential predictors for streamflow forecasting and proved its effectiveness for one- and multistep-ahead streamflow forecasting at two sites in Dongjiang basin, southern China. Also, Coulibaly [19] studied the potential of the echo state network (ESN) to make long-term predictions of lake water levels and applied the ESN to the Great Lakes. Coppola et al. [20] used an artificial neural network (ANN) to predict transient water levels in a complex multilayered groundwater system under various conditions. Wang et al. [21] used a support vector regression method to model the relationship between the water level of the lake and the amount of water released from the reservoir. All these methods have enabled progress in this research area; however, simulating a system as complicated as the relationship between a dam and lake levels remains undeniably challenging.

In this article, the deep learning method is proposed to address this problem. Different from other methods, deep learning networks allow computational models composed of multiple processing layers to learn representations of data with multiple levels of abstraction, and these networks simulate the way the human brain works [22]. The deep learning method has been applied in speech recognition, visual object recognition, object detection and many other domains, such as drug discovery and genomics, and has great potential in dealing with sequential data such as text and speech [23]. In this study, deep learning is used as an approach to establish a model that can portray the relationship between the water level of Dongting Lake and the influence of the TGD. With this model, it is feasible to predict the water level of Dongting Lake on a daily basis and shed light on the weight of the TGD influence and therefore provide a reference for flood control and the dam's operation.

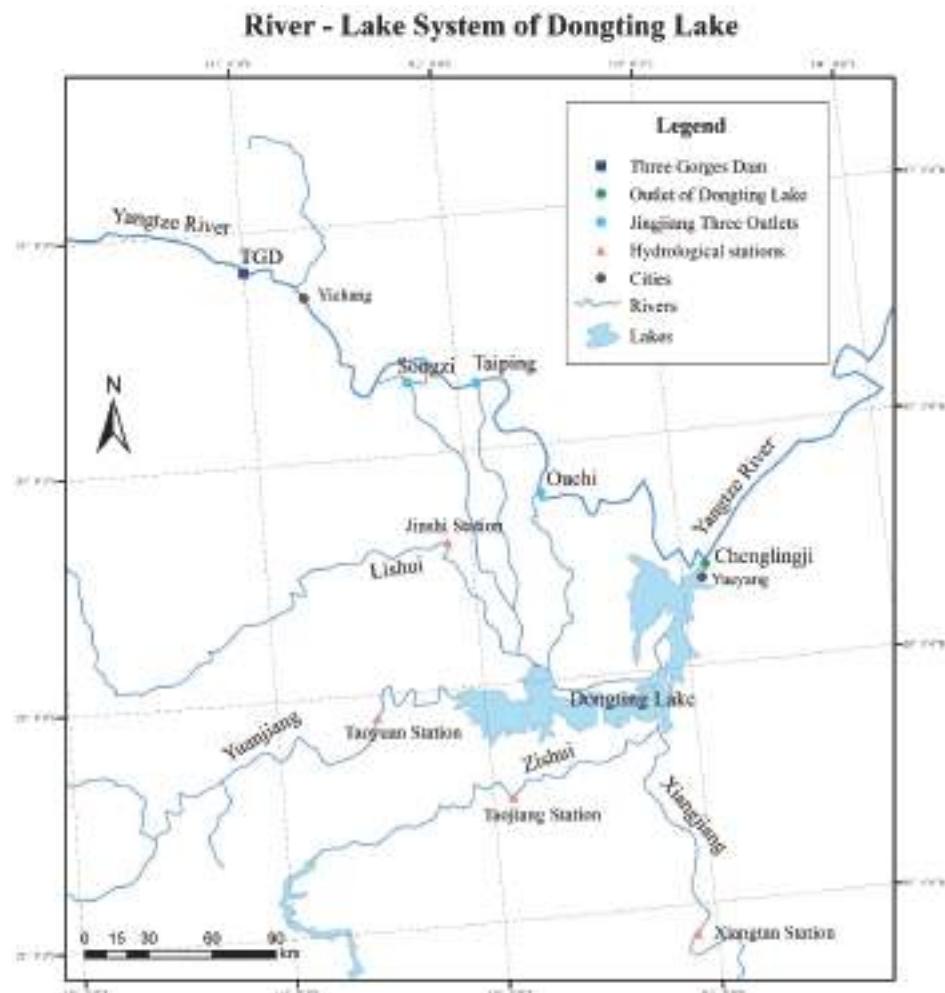
## 2. Materials and Methods

### 2.1. Study Area and Data

Dongting Lake is on the south bank of the Jingjiang River, which is another name for the Yangtze River in a specific segment in the middle reach of the Yangtze River region (approximately  $28^{\circ}30' \text{ N}$ – $30^{\circ}20' \text{ N}$ ,  $111^{\circ}40' \text{ E}$ – $113^{\circ}10' \text{ E}$ ) [24,25]. Dongting Lake is not only the landmark that divides the provinces of Hubei and Hunan, but also the second-largest freshwater lake in China. As shown in Figure 1, Dongting Lake is directly linked to the Yangtze River in the north, and the water is fed through three entrances: Songzi, Taiping, and Ouchi, which are often called the Jingjiang Three Outlets [26]. There are also four major rivers that drain into Dongting Lake to the south and west: the Xiangjiang, Zishui, Yuanjiang, and Lishui Rivers. The water in Dongting Lake flows back into the Yangtze River through Chenglingji in Yueyang. Dongting Lake can be divided into three parts: East Dongting, West Dongting, and South Dongting Lakes. Among these lakes, East Dongting Lake is the largest, which comprises 50% of the whole area [27]. Therefore, this lake is the main study object in most cases. The water level of Dongting Lake changes seasonally, and there can be a wide disparity between dry seasons and wet seasons. Generally, wet seasons occur from April to September with higher water levels, and October to March are considered dry seasons with relatively lower water levels [28].

The TGD is situated in the Xiling Gorge of the Yangtze River and is the world's largest hydraulic project with a normal pool level of 175 m after completion. The construction of the TGD was a 17-year, tri-phase project, and the dam is expected to withstand a 100-year flood [8]. On June 1, 2003, the impounding process was officially initiated, and the TGD was first put into use; from that point, the TGD has influenced the water flow in the lower reaches of the Yangtze River with main functions including flood control, power generation, shipping, and water supply. The regulation of water level in the TGD is based on flood peak reduction and drought flow recharge. Thus, during the flood season, the upstream flood peak is drastically reduced to alleviate the pressure of downstream flood prevention, and during the drought season, the discharge is supplemented to attempt to alleviate downstream drought conditions while increasing channel depth and improving the ecology [29]. The impoundment of the TGD is implemented every year approximately from September to October [30].

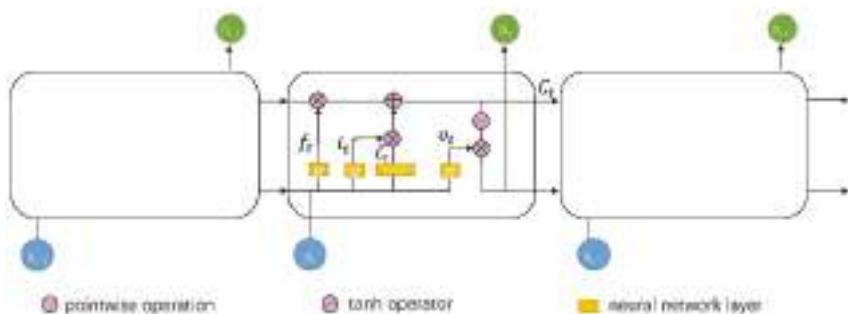
The data needed in this study for Dongting Lake were obtained from the website of the Hunan hydrology official network (<http://www.hnsw.com.cn/Default.aspx>), which provides data containing water level, water discharge, flood alert level, and others. The TGD data for this study were acquired from the China Three Gorges Corporation official website (<http://www.ctg.com.cn/sxjt/sqqk/index.html>), which chronicles in detail the daily discharge of the reservoir.



**Figure 1.** River-lake system of Dongting Lake.

## 2.2. Methodology

Recurrent neural networks, often known as RNNs, are networks which contain loops. As Fausett [31] said, the RNN is an ANN with arbitrary connections between neurons. The key point is that the recurrent connections allow a “memory” of previous inputs to persist in the network’s internal state and thereby influence the network output [32]. However, for standard RNN structures, the previous information often decays or blows up exponentially as it cycles around the network’s recurrent connections. This effect is often referred to in the literature as the vanishing gradient problem [33]. To solve this problem, Long Short-Term Memory networks (usually abbreviated to “LSTMs”) are introduced by Hochreiter and Schmidhuber [34]. LSTMs are a special type of RNN designed to avoid the long-term dependency problem, and the structure is illustrated in Figure 2. In Figure 2,  $x_t$  refers to the vector at time step  $t$ , which is the input vector, and  $h_t$  is the output hidden vector.  $h_t$  contains information from  $h_{t-1}$ , and together with the input vector at time  $t$  ( $x_t$ ), the information is passed on to the next time step  $t + 1$ , ensuring the information will persist.



**Figure 2.** Structure of the LSTM network.

Inside the LSTM cells, the most important concept is gates [34], which contain a sigmoid neural net layer and a pointwise multiplication operation and serve as a filter to optionally allow information through to protect and control the cell state. Typically, an LSTM has three of these gates: a forget gate, an input gate, and an output gate. The forget gate controls the information that is allowed through, whereas the input gate decides which values are used to update the information, and the output gate combines the results above and delivers a filtered output. The following Equations (1)–(6) demonstrate in detail how the LSTM cell maps an input vector sequence  $x$  to a hidden vector sequence  $h$ . In these equations,  $f_t$ ,  $i_t$ ,  $o_t$ , and  $C_t$  refer to the forget gate, input gate, output gate, and memory cell vectors, respectively, and  $W_f$ ,  $W_i$ ,  $W_o$ , and  $W_C$  are the weighted parameter matrices.  $\sigma$  and  $\tanh$  are the activation functions computed as Equations (7) and (8).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (8)$$

With each LSTM cell functioning in this manner and multiple LSTM cells stacked together, the LSTM network can constitute a complicated structure, which better serves the discovery of complex relationships between inputs and outputs [35]. Considering the structure of the collected data, we propose to consider this problem from the time series perspective. Because LSTMs have been proven to have better success in capturing long-term dependencies within a sequence [36], we suggest using the LSTM network to address the problem.

### 2.3. LSTM Model Establishment

#### 2.3.1. Variable Selection

The water level data of Dongting Lake are quintessential to this study. Because the water level change at Chenglingji station can reflect the changes in all of Dongting Lake's water levels [37],

the water level at Chenglingji station is used to represent the water level of Dongting Lake. Eleven years of daily data are extracted for this study, which covers the time period from 2003 to 2013 and is the amount of time the TGD has been actively used. The lake water level varies continuously, and this information is set to persist through the network. Many factors can contribute to changes in the water levels of Dongting Lake, because the lake is a very intricate system. In this study, we consider the water inflow and rainfall. As introduced in the study area section and shown in Figure 1, Dongting Lake is fed by the Yangtze River to the north and four other rivers to the south and west, which are the Xiangjiang, Zishui, Yuanjiang, and Lishui Rivers. The inflow of the Yangtze River is directly linked to the water discharge of the upstream TGD, whereas the inflow of the Xiangjiang, Zishui, Yuanjiang, and Lishui Rivers can be measured by the discharge data at the Xiangtan, Taojiang, Taoyuan, and Jinshi Stations, respectively. The rainfall at Dongting Lake also plays an important role in the water level variations in Dongting Lake and should be considered as a factor. The precipitation at Dongting Lake can be represented by the precipitation measured at the nearest weather station, which is the Yueyang weather station. Therefore, the daily precipitation data are obtained from the Yueyang weather station from 2003 to 2013. As a result, six factors are considered to determine the daily water level of Dongting Lake: the daily average TGD discharge, daily average discharge at Xiangtan Station, daily average discharge at Taojiang Station, daily average discharge at Taoyuan Station, daily average discharge at Jinshi Station, and daily average precipitation.

Quantitatively, a Grey relational analysis (GRA) was conducted to examine these six factors' correlation to the water level variation. GRA is a measure of the difference between data sequences and can be used to analyze the degree of correlation between these sequences [38]. Let the reference sequence and sequence for comparison be represented as  $x_o(k)$  and  $x_i(k)$ ,  $i = 1, 2, \dots, 6$ ;  $k = 1, 2, \dots, n$ , respectively, where  $n$  is the total number of observations. Here,  $x_o(k)$  is the water level sequence and  $x_i(k)$  is the sequence of the other factors, namely the TGD discharge sequence, Xiangtan discharge sequence, Taojiang discharge sequence, Taoyuan discharge sequence, Jinshi discharge sequence, and the precipitation sequence. The Grey relational grades are calculated following the procedures below [39]:

1. A series of various units must be transformed to be dimensionless using a normalization method. For instance, a comparability sequence  $x_i(k)$  is transformed as follows:

$$x_i^*(k) = \frac{x_i(k) - \min(x_i(k))}{\max(x_i(k)) - \min(x_i(k))} \quad (9)$$

In Equation (9),  $x_i^*(k)$  is the sequence after data preprocessing,  $\max(x_i(k))$  stands for the largest value of  $x_i(k)$ , whereas  $\min(x_i(k))$  stands for the smallest value of  $x_i(k)$ .

2. Calculate the Grey relational coefficients using the preprocessed sequences. The Grey relational coefficient is defined as below:

$$\gamma_i(k) = \frac{\min_{\forall j \in i} \min_{\forall k} |x_o^*(k) - x_i^*(k)| + \rho \max_{\forall j \in i} \max_{\forall k} |x_o^*(k) - x_i^*(k)|}{|x_o^*(k) - x_i^*(k)| + \rho \max_{\forall j \in i} \max_{\forall k} |x_o^*(k) - x_i^*(k)|} \quad (10)$$

In Equation (10),  $\rho$  is the distinguishing coefficient and normally  $\rho$  is set at 0.5.

3. Calculate the Grey relational grade  $r_i$ , which is the mean value of each Grey relational coefficient and is defined as follows:

$$r_i = \frac{1}{n} \sum_{k=1}^n \gamma_i(k) \quad (11)$$

The GRA results are listed in Table 1, showing correlations between each of the six factors and the water level variable, which is the target variable. From Table 1, we can see these factors have similar correlation with the target variable and the TGD discharge factor has the strongest correlation.

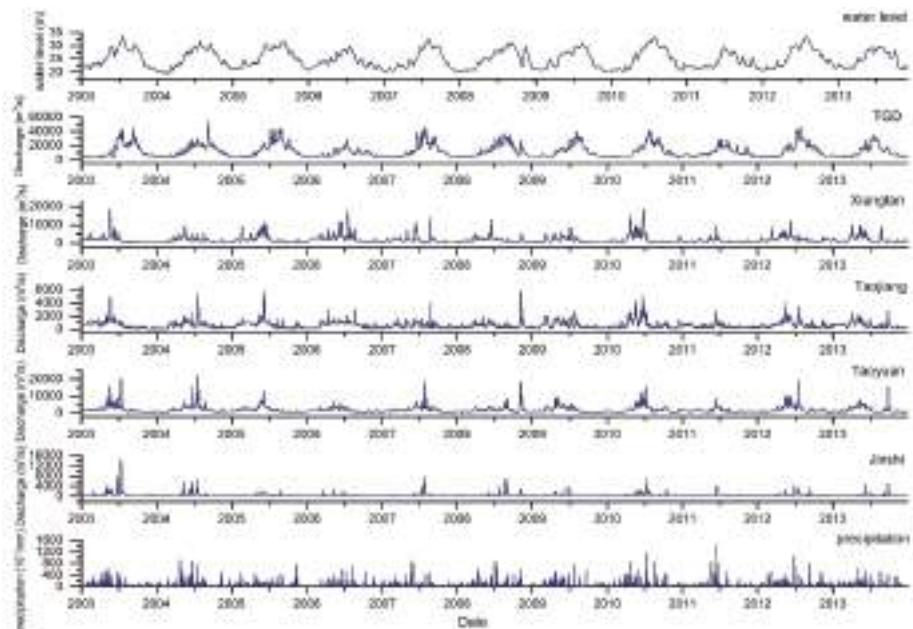
**Table 1.** GRA results.

Variables	TGD discharge	Xiangtan discharge	Taojiang discharge	Taoyuan discharge	Jinshi discharge	Precipitation
Grey relational grade	0.7242	0.6699	0.6798	0.6570	0.6140	0.6197

As a result, these six factors as well as the daily average water level data are chosen as the seven variables required for the LSTM network, with the water level variable being the target variable. The seven variables are listed in Table 2 (variable is shortened to var), and the values of the variables are shown in Figure 3, from which we can see a synchronized seasonal change.

**Table 2.** Selected variables.

Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7
Water level	TGD discharge	Xiangtan discharge	Taojiang discharge	Taoyuan discharge	Jinshi discharge	Precipitation

**Figure 3.** Data display of the selected variables.

### 2.3.2. Data Processing

Because LSTMs are sensitive to the scale of the input data, specifically when the sigmoid or tanh activation functions are used, it is preferable to scale the data before putting them into use. In this study, we scale the data using z-score standardization, which converts the dataset into a new dataset with a zero mean and unit variance [40]. The z-score is calculated using Equation (12), where  $\mu$  is the dataset mean and  $\sigma$  is the dataset standard deviation.

$$z = \frac{x - \mu}{\sigma} \quad (12)$$

Furthermore, the dataset contains the seven variables covering eleven years, and the time series data with multiple variables cannot be used directly for the LSTM network. The data must be restructured to a supervised learning dataset. Here, we apply the sliding window method, which uses the value at the previous time step to predict the value at the next time step. Thus, the data are reorganized, as shown in Table 3. The water level variable is the target variable, whose value at time  $t$  will be predicted based on the seven variables from time  $t - 1$ . Now, the problem is reframed to a supervised learning prediction problem.

**Table 3.** Converting variables to a supervised dataset for the LSTM network.

Var1 ( $t - 1$ ) <sup>1</sup>	Var2 ( $t - 1$ )	Var3 ( $t - 1$ )	Var4 ( $t - 1$ )	Var5 ( $t - 1$ )	Var6 ( $t - 1$ )	Var7 ( $t - 1$ )	Var1 ( $t$ )
Water level	TGD discharge	Xiangtan discharge	Taojiang discharge	Taoyuan discharge	Jinshi discharge	Precipitation	Water level

<sup>1</sup> Var1 ( $t - 1$ ) refers to variable 1 at time  $t - 1$ , and the same is true for the other variables.

Like other supervised learning problems, the dataset is split into a training dataset and a test dataset. Here, we use the first eight years of data (2003–2010) as the training data set and the last three years (2011–2013) as the test data set. Then, we split the training and test sets into input and output variables, with the last column from Table 3 being the output variable and the remainder being the input variables. Finally, the inputs are reshaped into the 3-D format expected by LSTMs, which include “samples, time-steps, and features”.

### 2.3.3. LSTM Network Design

The LSTM network design follows the “rough to fine” principle. That is, first, we come up with a base model with simple structures and default settings and then we tune the model with different hyperparameters. After that, we update the model, thus enhancing its skills, and finally we have the best model to forecast water levels.

#### Base Model Design

We designed the LSTM network with a hidden layer and an output layer with one neuron in each layer to predict the water level. The input shape will be one timestep with eight features, and the mean squared error (MSE) will be used for the loss function. The batch size and epoch hyperparameters are set at 1 and 100 respectively. The activation function is set as default, which is tanh.

A rolling forecast scenario is used for model construction, which means that each time step of the test dataset will be walked one at a time. The model will make a forecast for the current time step, then the actual expected value from the test set will be taken and made available to the model for the forecast on the next time step [41]. All forecasts on the test dataset will be collected and an error score will be calculated to summarize the performance of the model.

#### Model Evaluation

The root mean square error (RMSE) is used for evaluation because the RMSE finds large errors and results in a score that is in the same units as the forecast data. Also, the coefficient of determination ( $R^2$ ) is used to measure prediction performance. These two indexes are calculated using Equations (13) and (14) as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (13)$$

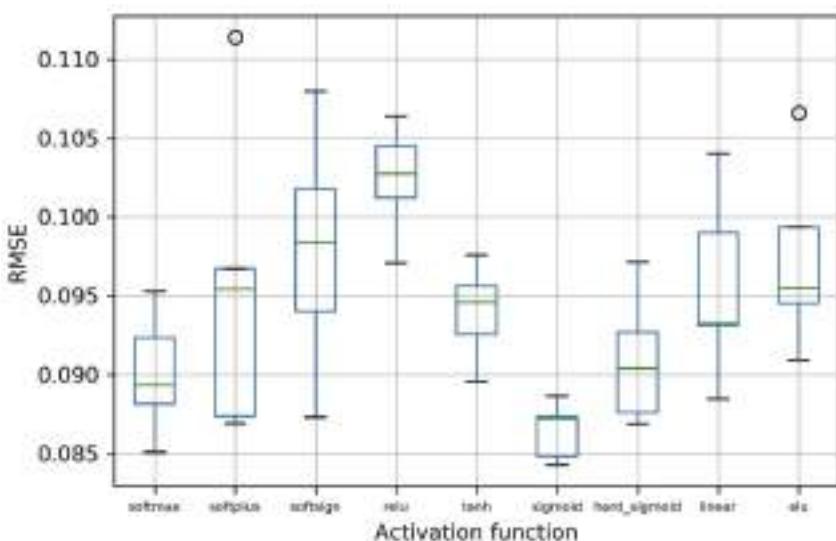
$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (14)$$

In the equations,  $n$  stands for the number of values,  $y_i$  refers to the actual water level,  $\hat{y}_i$  refers to the predicted water level, and  $\bar{y}_i$  stands for the mean actual water level. RMSE measures the average of the squares of the errors, so the closer the RMSE is to 0, the better. The coefficient of determination measures how well the regression predictions approximate the real data points, and if  $R^2$  is 1, it indicates that the predictions perfectly fit the data. That is, the closer  $R^2$  is to 1, the better the prediction model.

### Model Optimization

Model optimization involves tuning hyperparameters, such as epoch, batch size, neuron numbers, activation function, layers, and the optimization algorithm; a grid search method is implemented in this process. For epoch, batch size, and neuron numbers, the grid search follows the “rough to fine” principle, which means large strides are used in the first round of the grid search to find the most appropriate range and then small strides are used instead to home in on the best hyperparameters. A validation dataset is created by splitting 30% from the training dataset and the RMSE is used for evaluating the model performance. As a result, the hyperparameters of epoch, batch size, and neuron numbers were set at 200, 32, and 13 for the LSTM network respectively.

For the activation function, as introduced in Section 2.2, the LSTM network involves two kinds of activation functions: the sigmoid activation function and the tanh activation function. The sigmoid function is associated with “gates,” which is the core structure of the LSTM network, so it cannot be changed. However, the tanh function is used for data output and it can be replaced. In this case, a wide range of functions are considered, which include “softmax,” “softplus,” “softsign,” “ReLU,” “tanh,” “sigmoid,” “hard sigmoid,” “linear,” and “ELU” (exponential linear unit). Each of the functions will be tested for the LSTM network, and in our case, each were run 10 times given that the random initial conditions for an LSTM network can result in very different results each time a given configuration is trained. Each time, the RMSE error will be recorded, and the results will all be illustrated in a box-and-whisker plot shown in Figure 4. From this plot, we can see that the activation function “sigmoid” had the best overall performance and hence was chosen as the activation function for the LSTM network. The sigmoid activation function is defined by Equation (7).

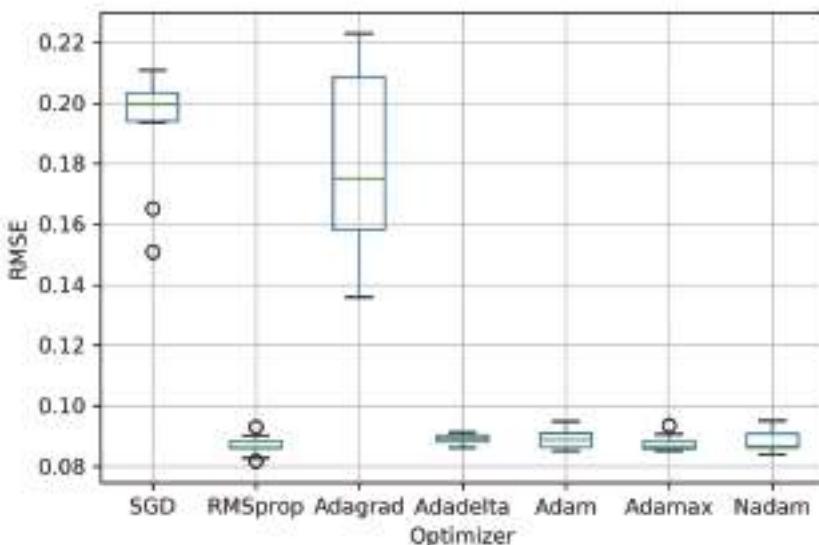


**Figure 4.** Box-and-whisker plot for activation function selection.

In training, the loss function lets us quantify the quality of any particular set of weights. The goal of optimization is to find the weights that minimize the loss function. Gradient descent is a commonly used optimization algorithm for finding the minimum of a function. To minimize a loss function  $J(\theta)$  using gradient descent, it updates the parameters in the opposite direction of the gradient at the current point. The learning rate  $\eta$  determines the size of the steps to take to reach a (local) minimum. Stochastic gradient descent (SGD) is a fundamental gradient descent algorithm that performs a parameter update for each training example  $x^{(i)}$  and  $y^{(i)}$  as follows [42]:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (15)$$

In Equation (15),  $\theta$  stands for the model's parameters, and  $\nabla_{\theta} J(\theta; x^{(i)}; y^{(i)})$  is the gradient at the current example. There are also a lot of variations of gradient descent algorithm. For example, Momentum is a method that helps accelerate SGD in the relevant direction and Nesterov accelerated gradient (NAG) adjusts the gradient descent direction. Adagrad is an algorithm that adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters. Adadelta is an extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Root mean square prop (RMSprop) is an algorithm proposed by Geoff Hinton [42], which is also an extension of Adagrad, that aims to solve the rapid decreasing learning rate problem. Adaptive moment estimation (Adam) is another method that combines RMSprop and momentum. Additionally, the AdaMax method is a variation of Adam, and Nadam is the combination of NAG and Adam. We explored the applicability of different optimization algorithms including SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, and Nadam. The RMSEs of each algorithm exerted on the LSTM network are calculated and shown in Figure 5. From this box-and-whisker plot, we can see that aside from SGD and Adagrad, the other four algorithms had similar performances, but RMSprop has the potential of reaching the lowest RMSE. As a result, RMSprop was chosen as the optimization algorithm for the model.



**Figure 5.** Box-and-whisker plot for optimization algorithm selection.

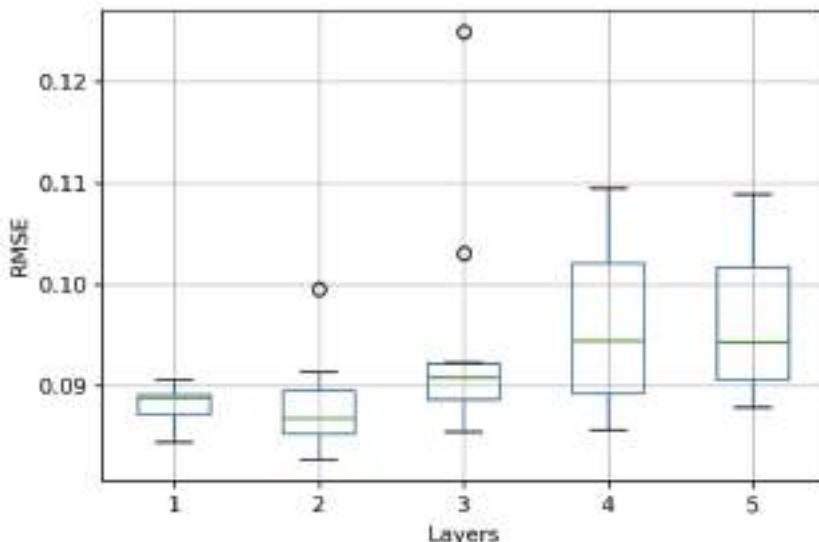
The most important difference between RMSprop and SGD is that RMSprop keeps a moving average of the squared gradient for each weight. Let  $g_t$  be the gradient of the objective function at the current parameter at time step  $t$ . The RMSprop algorithm updates parameters as follows [42]:

$$\mathbb{E}[g^2]_t = \gamma \mathbb{E}[g^2]_{t-1} + (1 - \gamma) g_t^2 \quad (16)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_t + \epsilon}} g_t \quad (17)$$

In Equations (16) and (17),  $\gamma$  is the decay rate, which is normally set at 0.9, and  $\epsilon$  is a smoothing term that avoids division by zero (often set at  $1 \times 10^{-6}$ ). Also, a good default value for the learning rate  $\eta$  is 0.001.

As for different numbers of hidden layers, the base model uses one LSTM layer and so we explore the possibility of multiple layers. We stack two LSTM layers together as the two-layer structure with 13 neurons on the first layer and half on the second layer. Similarly, we stack three LSTM layers together as the three-layer structures with 13 neurons on the first layer and two-thirds the number of neurons on the second layer and one-third on the third layer. Other multi-layer structures are designed the same way and each model with different layers is evaluated through experiment. The RMSE results are shown in Figure 6. From the figure, it can be seen that the two-layer structure had significantly better performance. Hence, we updated the model with two LSTM layers.



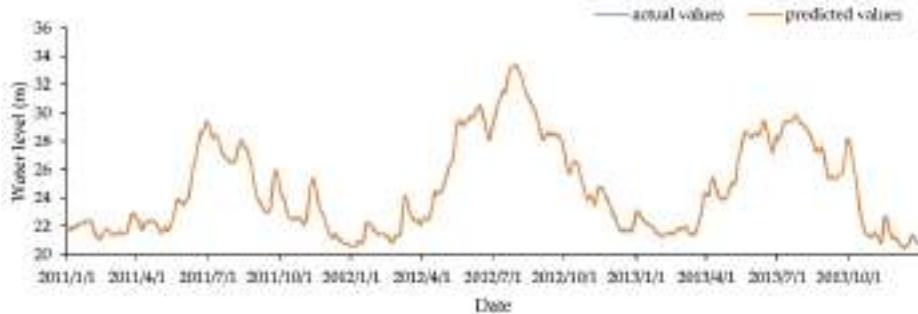
**Figure 6.** Box-and-whisker plot for layer number selection.

Piecing all the hyperparameters together, the LSTM network has been optimized and updated, thus the LSTM model for water level forecast is completed.

### 3. Results

We used the eight-year dataset (from 2003 to 2010) to train the LSTM network and then used this trained model to predict the daily water level for the next three years of data (from 2011 to 2013), thus testing the performance of the model. The experiment was carried out using Keras (version: 2.1.3) with the Tensorflow backend (version: 1.4.0). Keras is a high-level neural networks API developed

and maintained by Google engineer François Chollet, and TensorFlow is an open-source symbolic tensor manipulation framework developed by Google. The result is shown in Figure 7. From the figure, we can see that the actual value line and predicted value line were nearly coincident.



**Figure 7.** Forecasting results of the Dongting Lake water level using the LSTM model.

Quantitatively, with forecasts and actual values both inverted to their original scale, the RMSE and coefficient of determination ( $R^2$ ) were calculated to evaluate the model. The result showed the RMSE was 0.083, and the coefficient of determination was 0.999, which showed high precision. Because one of the main characteristics of deep learning is that it is stochastic, the model is very likely to obtain a different result every time we run the network. However, after a great deal of tests, the RMSE was always within the range of 0.080–0.100. The coefficient of determination stayed at 0.999, which showed the robustness of the LSTM network. Table 4 shows the results of running the model 10 times on the test dataset. The test results suggest that this deep learning network is a sound prediction model and capable of portraying the relationship between the TGD and Dongting Lake water level.

**Table 4.** Test results of running the LSTM model 10 times on the test dataset.

No.	RMSE	$R^2$
1	0.083	0.999
2	0.091	0.999
3	0.099	0.999
4	0.090	0.999
5	0.086	0.999
6	0.086	0.999
7	0.085	0.999
8	0.085	0.999
9	0.088	0.999
10	0.087	0.999

## 4. Discussion

### 4.1. Support Vector Machine Comparison Experiment

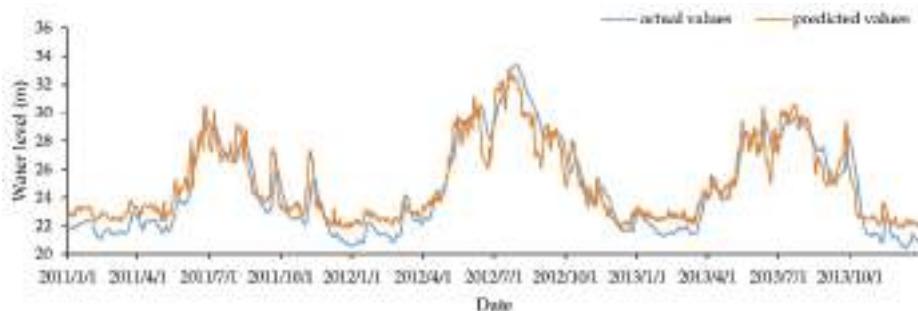
For comparison, we used a support vector machine (SVM) method to solve the same problem. SVMs are developed based on statistical learning theory and employ the structural risk minimization (SRM) principle for a global solution [43]. SVMs have demonstrated good performance in regression [44] and time series forecasting and prediction [45,46]. For these reasons, an SVM model was chosen for a comparison experiment. We built the SVM model based on the idea proposed by Wang et al. [21]. We selected the discharge of the TGD, Xiangtan, Taoyuan, Jinshi, and Taojiang, as well as the precipitation, as the inputs and consider the water level as the target variable. We used these

input factors to predict the water level of Dongting Lake, turning this subject into an SVM regression problem. The input and output variables are shown in Table 5.

**Table 5.** Input and output variables for the SVM model.

Input						Output
Var1	Var2	Var3	Var4	Var5	Var6	Target variable
TGD discharge	Xiangtan discharge	Taojiang discharge	Taoyuan discharge	Jinshi discharge	Precipitation	Water level

The training and test datasets were divided in the same manner as before, with the first eight years of data (2003–2010) used as the training dataset and the last three years of data (2011–2013) as the test dataset. The data were also scaled before being inputted into the SVM model. For the SVM model, we chose the epsilon-SVR model with the Radial Basis Function (RBF) being the kernel function. Other parameters for the model were optimized through the genetic algorithm used on the training dataset. The genetic algorithm (GA) was developed by Holland. The concept is based on the survival of the fittest [47]. It uses genes and chromosomes to represent variables and solutions and imitates the biological process to select the best one. Furthermore, as a result, the cost ( $c$ ), gamma ( $\gamma$ ), and epsilon ( $\epsilon$ ) parameters were set at 1.175, 11.239, and 0.035, respectively. Then, the established SVM model was used on the test dataset for prediction, and the result is illustrated in Figure 8 below.

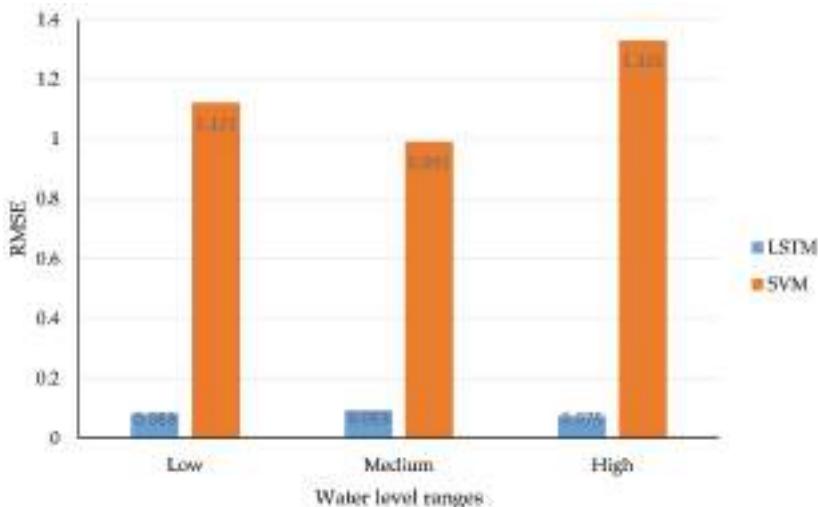


**Figure 8.** Forecasting results of Dongting Lake water levels using the SVM model.

From Figures 7 and 8, we see that the prediction deviation using the LSTM model was far less than when using the SVM model. Quantitatively, the calculated RMSE of the SVM model was 1.157, which was much larger than the LSTM model, and the  $R^2$  was 0.873, which was well under the 0.999 LSTM model performance.

Additionally, Figures 7 and 8 indicate that the SVM model tended to make higher predictions during low water periods and lower predictions during high water periods; whereas the LSTM model had an overall satisfying performance. Statistically, we analyzed the water level distribution from the test dataset through percentile and split the data at 33.3% and 66.7%, thus separating the water level data into low, medium, and high categories [18]. Then we calculated the RMSE of the actual water level and the predicted values in each category for both models; the results are shown in Figure 9.

Figure 9 shows that the LSTM model had relatively much smaller RMSEs in all three water level ranges. Additionally, the LSTM model had the best performance for high water levels and the RMSE difference between these three ranges was minute. In contrast, the SVM model had much larger RMSEs in all ranges and especially for low water levels and high water levels. In summary, the LSTM model could deliver better results and was more suitable for predicting the water level of Dongting Lake.

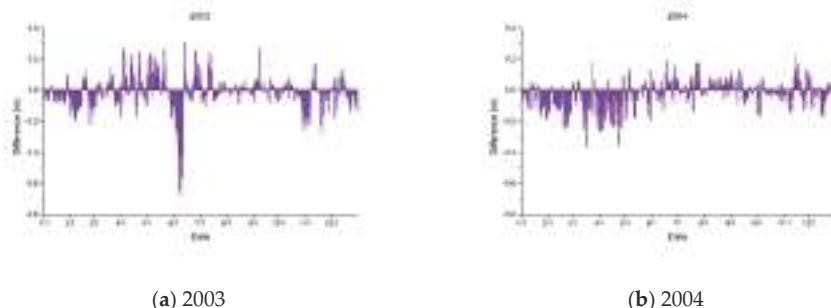


**Figure 9.** Model performances in different water level ranges.

#### 4.2. TGD's Impact on Dongting Lake Water Level

The LSTM network above has already captured the relationship of the TGD discharge and the water level of Dongting Lake. That is, given a certain amount of TGD discharge and other variables, we were able to predict the water level. This finding can shed some light on how the discharge of the TGD can affect the water level of Dongting Lake and provide a reference for water dispatch and flood control plans.

If we changed the TGD discharge variable and keep the other variables fixed, we could separate the TGD discharge impact from the complicated system and determine how it influenced the water level of Dongting Lake. We replaced the TGD discharge variable with the dam's daily inflow data for the purpose of simulating a situation where the TGD did not exist and determined how the water level of Dongting Lake changed accordingly [48]. We called this the mock data. The mock data included the inflow data of the TGD from 2003 to 2013, and the other variables were the same as before. We put the mock data into the LSTM network and obtained predictions for the water level of Dongting Lake from 2003 to 2013, and we called this result the mock water level. The difference was evaluated by subtracting the actual water level from the mock water level, which is shown in Figure 10.



**Figure 10. Cont.**

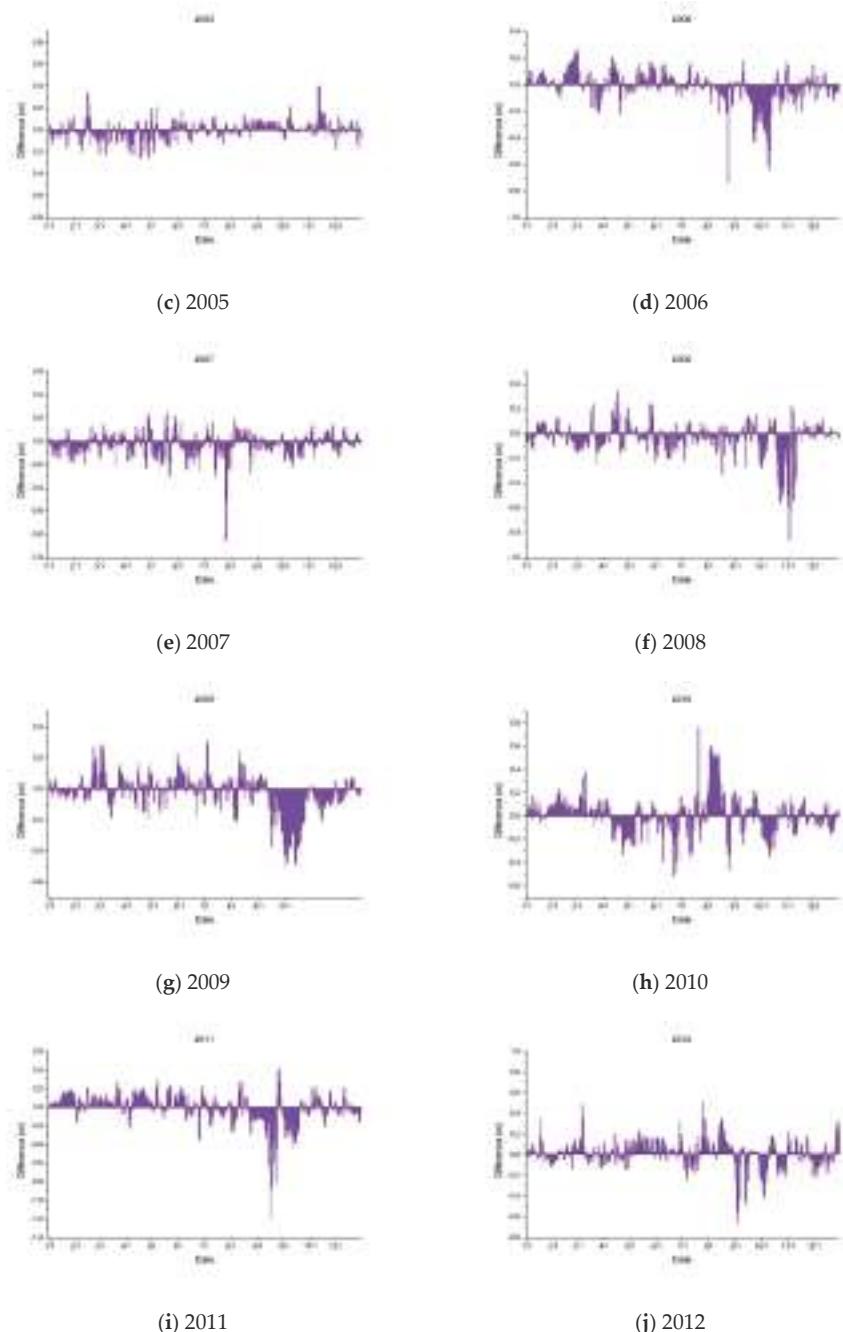
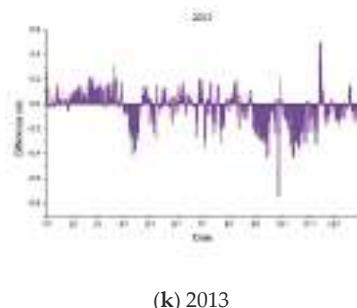


Figure 10. Cont.



(k) 2013

**Figure 10.** Water level difference between reality and simulation every year with (a–k) showing years 2003–2013, respectively.

In 2003, as shown in Figure 10a, large disparities appear in June. The water level dropped significantly in June compared to the mock data. This was because the TGD started its first impoundment in June 2003 until the water level of the dam reached 135 m at the end of June [49]. That time is when the TGD began to be actively used and started its first phase of operation. Based on this simulation experiment, the water level of Dongting Lake dropped 0.31 m on average during the TGD impoundment in June, and the largest drop on a single day was up to 0.65 m. There was also a noticeable drop in the water levels of Dongting Lake in November 2003 because the TGD impounded water again at that time, which enabled the dam water level to reach 139 m.

Through 2004 and 2005, the TGD did not operate regularly, so the impact on the water level variation was not large. Figure 10b,c shows that the water level of Dongting Lake dropped in the first half of the year in 2004 and 2005, and the water level rose slightly in the second half of the year in 2005.

In September 2006, the TGD started its second phase of operation, pushing the dam water level to 156 m. The impounding process lasted from September until the end of October, during which time there was a major drop in the water level of Dongting Lake, as shown in Figure 10d. The water level of Dongting Lake dropped 0.26 m on average according to the simulation experiment, and on extreme occasions, the drop was as much as 0.66 m.

In 2007, as illustrated in Figure 10e, the water level of Dongting Lake had a relatively large drop in July and August due to the flood control dispatch of the TGD.

In 2008, as shown in Figure 10f, there was a sharp drop in the water level in October and November. The reason for this decline is that the TGD implemented its third phase of operation, and the impounding process completed the goal of reaching the 175 m reservoir water level. Accordingly, the Dongting Lake water level dropped by 0.34 m on average during that time, and the largest drop in a single day was as much as 0.86 m.

In 2009, there was a relatively large increase around March as a result of the TGD's water replenishment strategy during dry seasons. The water level of Dongting Lake showed a substantial drop in October and November because of the impoundment of the TGD (see Figure 10g).

In 2010, as we can see in Figure 10h, the water level variation was different from before, and what makes this year special was that the Yangtze River experienced one of the largest floods during summer. The flood peak in July was the third largest flood peak ever recorded in the Yangtze River hydrological record, and the flood reached more than  $70,000 \text{ m}^3/\text{s}$  flow per second [50]. Nevertheless, the water level of Dongting Lake did not increase much as a result. In fact, the lake level dropped 0.22 m after the TGD discharge was reduced to  $30000 \text{ m}^3/\text{s}$  (maximum outflow), as opposed to  $70000 \text{ m}^3/\text{s}$  (maximum inflow) if the TGD does not exist. The water level increased substantially in August after the flood, which was when the TGD opened sluices to release the flood water. By doing so, the TGD managed to stagger the flood peak to alleviate the rise in water level in Dongting Lake. Then, the water level showed a moderate decline in September and November due to the impoundment of the TGD.

In 2011, there was a mild increase in the water level for the first half of the year because of the dispatch of TGD water replenishment, which started on December 29, 2010, and ended on June 10, 2011. The replenishment raised the water level by 0.16 m on average, and the largest increase reached 0.3 m in the experiment (see Figure 10i). There was a drastic drop in September and October because the TGD was impounding water in order for the dam water level to reach 175 m again, and the drop was as much as 1.2 m.

In 2012, the water level of Dongting Lake increased slightly from January to June due to TGD water replenishment, and the peak increase was 0.48 m (see Figure 10j). Similar to 2010, the Yangtze River observed a large flood again in July, but the water level of Dongting Lake declined by 0.15 m during the first flood peak. Then, the water level rose considerably in August when the TGD opened the sluice and released the flood water. The water level dropped again in the fall because of the TGD impoundment, and the dam water level reached 175 m again on October 30.

In 2013, the water level of Dongting Lake showed a steady increase in the first quarter of the year with an average increase of 0.15 m due to the TGD water replenishment, and the lake showed some mild, intermittent decreases during wet seasons (see Figure 10k). There was a major decline from September to November when the TGD entered its impounding period until the reservoir water level again reached 175 m, and the largest drop of 0.75 m occurred on a single day.

## 5. Conclusions

In this article, a deep learning model based on an LSTM network was proposed to predict the daily water level of Dongting Lake. Seven variables were selected as input factors quintessential to the water level variations, and eight years of daily data (from 2003 to 2012) were used to train the model. Then, the model was tested on the daily data for the next three years (from 2011 to 2013). The experiments showed that the LSTM model predicted the water level of Dongting Lake with high precision and delivered better results compared to an SVM model.

This LSTM prediction model also established the connections between the water level of Dongting Lake and the TGD discharge. Eleven years of mock data (from 2003 to 2013) were used to simulate a situation where the TGD does not exist to examine how the water level of Dongting Lake differed from reality. Through this experiment, we made the following conclusions:

1. The water level of Dongting Lake dropped conspicuously when the TGD is being impounded, which occurred annually from September to November. The drop was approximately 0.3 m on average and could be as large as 1.2 m in a single day.
2. The water level increased mildly during dry seasons because of the TGD water replenishment strategy, which demonstrated the water conservancy effects of the dam.
3. There was a decline in the water level of Dongting Lake during flood seasons (mostly July during flood peaks) and a lagged increase occurred later, proving that the dam's effects on flood control and staggering the flood peak.

Overall, the TGD discharge control and its water regulation plans had a strong impact on the water level variation of Dongting Lake. With the forecasting model proposed in this article, the daily water level of Dongting Lake could be predicted, which will help with flood control and water regulation.

**Author Contributions:** C.L. collected data, and conceived and designed the study with the support of Q.D. C.L., H.L., M.L. and Q.D. drafted and revised the article together. All authors read and approved the final manuscript.

**Funding:** This research was funded by the National Key Research and Development Program of China (2016YFC0803106).

**Acknowledgments:** The authors would like to thank Qingyun Du from Wuhan University for his valuable suggestions. This work was supported by the National Key R&D Program of China (2016YFC0803106).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Beeton, A.M. Large freshwater lakes: Present state, trends, and future. *Environ. Conserv.* **2002**, *29*, 21–38. [[CrossRef](#)]
- Yuan, Y.; Zhang, C.; Zeng, G.; Liang, J.; Guo, S.; Huang, L.; Wu, H.; Hua, S. Quantitative assessment of the contribution of climate variability and human activity to streamflow alteration in Dongting Lake, China. *Hydrolog. Processes* **2016**, *30*, 1929–1939. [[CrossRef](#)]
- Mitsch, W.J.; Gosselink, J.G. *Wetlands*, 4th ed.; John Wiley & Sons Ltd.: New York, NY, USA, 2007.
- Yuan, Y.; Zeng, G.; Liang, J.; Huang, L.; Hua, S.; Li, F.; Zhu, Y.; Wu, H.; Liu, J.; He, X. Variation of water level in Dongting Lake over a 50-year period: Implications for the impacts of anthropogenic and climatic factors. *J. Hydrol.* **2015**, *525*, 450–456. [[CrossRef](#)]
- Hayashi, S.; Murakami, S.; Xu, K.-Q.; Watanabe, M. Effect of the Three Gorges Dam Project on flood control in the Dongting Lake area, China, in a 1998-type flood. *J. Hydro-Environ. Res.* **2008**, *2*, 148–163. [[CrossRef](#)]
- Jiang, L.; Ban, X.; Wang, X.; Cai, X. Assessment of hydrologic alterations caused by the Three Gorges Dam in the middle and lower reaches of Yangtze River, China. *Water* **2014**, *6*, 1419–1434. [[CrossRef](#)]
- Nakayama, T.; Shankman, D. Impact of the Three-Gorges Dam and water transfer project on Changjiang floods. *Glob. Planet. Chang.* **2013**, *100*, 38–50. [[CrossRef](#)]
- Guo, H.; Hu, Q.; Zhang, Q.; Feng, S. Effects of the Three Gorges Dam on Yangtze River flow and river interaction with Poyang Lake, China: 2003–2008. *J. Hydrol.* **2012**, *416*–417, 19–27. [[CrossRef](#)]
- Hu, Q.; Feng, S.; Guo, H.; Chen, G.; Jiang, T. Interactions of the Yangtze river flow and hydrologic processes of the Poyang Lake, China. *J. Hydrol.* **2007**, *347*, 90–100. [[CrossRef](#)]
- Zhou, J.; Peng, T.; Zhang, C.; Sun, N. Data pre-analysis and ensemble of various artificial neural networks for monthly streamflow forecasting. *Water* **2018**, *10*, 628. [[CrossRef](#)]
- Yu, Y.; Zhang, H.; Singh, V. Forward prediction of runoff data in data-scarce basins with an improved Ensemble Empirical Mode Decomposition (EEMD) model. *Water* **2018**, *10*, 388. [[CrossRef](#)]
- Delgado-Ramos, F.; Hervás-Gámez, C. Simple and low-cost procedure for monthly and yearly streamflow forecasts during the current hydrological year. *Water* **2018**, *10*, 1038. [[CrossRef](#)]
- Lai, X.J.; Jiang, J.H.; Huang, Q. Effect of Three Gorge Reservoir on the water regime of the Dongting Lake during important regulation periods. *Resour. Environ. Yangtze Basin* **2011**, *20*, 167–172.
- Wu, N.; Luo, Y.; Liu, T.; Huang, Z. Experimental study on the effect of the Three Gorges Project on water level in Lake Poyang. *J. Lake Sci.* **2014**, *26*, 522–528. [[CrossRef](#)]
- Jiang, J.; Huang, Q. Study on impacts of the Three Gorge Project on water level of Dongting Lake. *Resour. Environ. Yangtze Basin* **1996**, *5*, 367–374.
- Liu, Z.; Zhou, P.; Chen, X.; Guan, Y. A multivariate conditional model for streamflow prediction and spatial precipitation refinement. *J. Geophys. Res. Atmos.* **2015**, *120*, 10116–10129. [[CrossRef](#)]
- Khedun, C.P.; Mishra, A.K.; Singh, V.P.; Giardino, J.R. A copula-based precipitation forecasting model: Investigating the interdecadal modulation of ENSO’s impacts on monthly precipitation. *Water Resour. Res.* **2014**, *50*, 580–600. [[CrossRef](#)]
- Liu, Z.; Zhou, P.; Zhang, Y. A Probabilistic Wavelet-Support Vector Regression Model for Streamflow Forecasting with Rainfall and Climate Information Input. *J. Hydrometeorol.* **2015**, *16*, 2209–2229. [[CrossRef](#)]
- Coulibaly, P. Reservoir Computing approach to Great Lakes water level forecasting. *J. Hydrol.* **2010**, *381*, 76–88. [[CrossRef](#)]
- Coppola, E.; Szidarovszky, F.; Poulton, M.; Charles, E. Artificial neural network approach for predicting transient water levels in a multilayered groundwater system under variable state, pumping, and climate conditions. *J. Hydrol. Eng.* **2003**, *8*, 348–360. [[CrossRef](#)]
- Wang, M.; Dai, L.; Dai, H.; Mao, J.; Liang, L. Support vector regression based model for predicting water level of Dongting Lake. *J. Drain. Irrig. Mach. Eng.* **2017**, *35*, 954–961. [[CrossRef](#)]
- Chen, Y.; Fan, R.; Yang, X.; Wang, J.; Latif, A. Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning. *Water* **2018**, *10*, 585. [[CrossRef](#)]
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
- Ding, X.; Li, X. Monitoring of the water-area variations of Lake Dongting in China with ENVISAT ASAR images. *Int. J. Appl. Earth Obs. Geoinf.* **2011**, *13*, 894–901. [[CrossRef](#)]

25. Li, F.; Huang, J.; Zeng, G.; Yuan, X.; Li, X.; Liang, J.; Wang, X.; Tang, X.; Bai, B. Spatial risk assessment and sources identification of heavy metals in surface sediments from the Dongting Lake, Middle China. *J. Geochem. Explor.* **2013**, *132*, 75–83. [[CrossRef](#)]
26. Li, Y.; Yang, G.; Li, B.; Wan, R.; Duan, W.; He, Z. Quantifying the effects of channel change on the discharge diversion of Jingjiang Three Outlets after the operation of the Three Gorges Dam. *Hydrol. Res.* **2016**, *47*. [[CrossRef](#)]
27. Qin, H. *The Succession and Reason Analysis of Hydrological Environment in the Dongting Lake in Recent 50 Years*; Hunan Agricultural University: Changsha, China, 2013.
28. Shi, X.; Xiao, W.; Wang, Y.; Wang, X. Characteristics and factors of water level variations in the Dongting Lake during the recent 50 years. *South-to-North Water Transf. Water Sci. Technol.* **2012**, *10*, 18–22. [[CrossRef](#)]
29. Yang, Y.; Zhang, M.; Zhu, L.; Liu, W.; Han, J.; Yang, Y. Influence of large reservoir operation on water-levels and flows in reaches below dam: Case study of the Three Gorges Reservoir. *Sci. Rep.* **2017**, *7*, 15640. [[CrossRef](#)] [[PubMed](#)]
30. Zhang, J.; Feng, L.; Chen, L.; Wang, D.; Dai, M.; Xu, W.; Yan, T. Water compensation and its implication of the Three Gorges Reservoir for the river-lake system in the middle Yangtze River, China. *Water* **2018**, *10*, 1011. [[CrossRef](#)]
31. Fausett, L. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*; Prentice-Hall: Upper Saddle River, NJ, USA, 1994.
32. Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany; London, UK, 2012.
33. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [[CrossRef](#)] [[PubMed](#)]
34. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
35. Ke, J.; Zheng, H.; Yang, H.; Chen, X. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transp. Res. C-Emerg.* **2017**, *85*, 591–608. [[CrossRef](#)]
36. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6730–6739. [[CrossRef](#)]
37. Han, Q.; Zhang, S.; Huang, G.; Zhang, R. Analysis of long-term water level variation in Dongting Lake, China. *Water* **2016**, *8*, 306. [[CrossRef](#)]
38. Arce, M.E.; Saavedra, Á.; Míguez, J.L.; Granada, E. The use of grey-based methods in multi-criteria decision analysis for the evaluation of sustainable energy systems: A review. *Renew. Sustain. Energy Rev.* **2015**, *47*, 924–932. [[CrossRef](#)]
39. Tzeng, C.-J.; Lin, Y.-H.; Yang, Y.-K.; Jeng, M.-C. Optimization of turning operations with multiple performance characteristics using the Taguchi method and Grey relational analysis. *J. Mater. Process. Technol.* **2009**, *209*, 2753–2759. [[CrossRef](#)]
40. Kreyszig, E. *Advanced Engineering Mathematics*, 4th ed.; Wiley: New York, NY, USA, 1979.
41. Huang, L.; Hung, S.; Leu, J. A Study on the Effects of Rolling Forecast with Different Information Sharing Models on Supply Chain Performance. In Proceedings of the 40th International Conference on Computers & Industrial Engineering, Awaji Island, Japan, 25–28 July 2010; pp. 1–6.
42. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv*, 2016; arXiv:1609.04747.
43. Behzad, M.; Asghari, K.; Coppola, E.A. Comparative study of SVMs and ANNs in aquifer water level prediction. *J. Comput. Civ. Eng.* **2010**, *24*, 408–413. [[CrossRef](#)]
44. Dibike, Y.B.; Velickov, S.; Solomatine, D.; Abbott, M.B. Model induction with support vector machines: Introduction and applications. *J. Comput. Civ. Eng.* **2001**, *15*, 208–216. [[CrossRef](#)]
45. Kim, K.-J. Financial time series forecasting using support vector machines. *Neurocomputing* **2003**, *55*, 307–319. [[CrossRef](#)]
46. Thissen, U.; van Brakel, R.; de Weijer, A.P.; Melssen, W.J.; Buydens, L.M.C. Using support vector machines for time series prediction. *Chemometrics Intell. Lab. Syst.* **2003**, *69*, 35–49. [[CrossRef](#)]
47. Tayfur, G.; Singh, V.; Moramarco, T.; Barbetta, S. Flood hydrograph prediction using machine learning methods. *Water* **2018**, *10*, 968. [[CrossRef](#)]

48. Huang, Q.; Sun, Z.; Jiang, J. Impacts of the operation of the Three Gorges Reservoir on the lake water level of Lake Dongting. *J. Lake Sci.* **2011**, *23*, 424–428. [[CrossRef](#)]
49. Gao, B.; Yang, D.; Yang, H. Impact of the Three Gorges Dam on flow regime in the middle and lower Yangtze River. *Quat. Int.* **2013**, *304*, 43–50. [[CrossRef](#)]
50. Lai, X.-J.; Wang, Z.-M. Flood management of Dongting Lake after operation of Three Gorges Dam. *Water Sci. Eng.* **2017**, *10*, 303–310. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Multi-Objective Parameter Estimation of Improved Muskingum Model by Wolf Pack Algorithm and Its Application in Upper Hanjiang River, China

Tao Bai <sup>1,\*</sup>, Jian Wei <sup>1</sup>, Wangwang Yang <sup>2</sup> and Qiang Huang <sup>1</sup>

<sup>1</sup> State Key Laboratory of Eco-hydraulics in Northwest Arid Region of China, Xi'an University of Technology, Xi'an 710048, China; wjxaut@163.com (J.W.); wreshxaut@163.com (Q.H.)

<sup>2</sup> CCCC (Tianjin) Eco-Environmental Protection Design & Research Institute Co., Ltd., Tianjin 300461, China; yangwanghdj@126.com

\* Correspondence: baitao@xaut.edu.cn; Tel.: +86-029-8231-2036

Received: 31 August 2018; Accepted: 6 October 2018; Published: 10 October 2018

**Abstract:** In order to overcome the problems in the parameter estimation of the Muskingum model, this paper introduces a new swarm intelligence optimization algorithm—Wolf Pack Algorithm (WPA). A new multi-objective function is designed by considering the weighted sum of absolute difference (SAD) and determination coefficient of the flood process. The WPA, its solving steps of calibration, and the model parameters are designed emphatically based on the basic principle of the algorithm. The performance of this algorithm is compared to the Trial Algorithm (TA) and Particle Swarm Optimization (PSO). Results of the application of these approaches with actual data from the downstream of Ankang River in Hanjiang River indicate that the WPA has a higher precision than other techniques and, thus, the WPA is an efficient alternative technique to estimate the parameters of the Muskingum model. The research results provide a new method for the parameter estimation of the Muskingum model, which is of great practical significance to improving the accuracy of river channel flood routing.

**Keywords:** Muskingum model; wolf pack algorithm; parameters; optimization; flood routing

## 1. Introduction

In recent years, frequent floods have caused serious losses to people's lives and property [1]. Therefore, it is urgent to identify the flood routing rules to reduce flood losses. As a classic solution of flood routing, the Muskingum model continues to be a popular method for flood routing [2]. The Muskingum model was first developed by McCarthy [3]. In practical application, the key problem for applying the Muskingum model is parameter estimation [4,5], which is a highly nonlinear optimization problem. The studies on the Muskingum model mainly include the parameter estimation method and model improvement.

Traditional parameter estimation methods mainly include the Trial Algorithm (TA), the moment method, the least squares method, and the differential algorithm [5,6]. However, these methods are limited by the optimal estimation of the channel storage curve, which has led to a significant error between the calculated results and the observed data. In recent years, intelligence algorithms have been widely used in solving the nonlinear problem accurately and efficiently [7]. Many researchers have applied various techniques to estimate the parameters of the Muskingum model in recent years. For example, Luo and Xie [4] applied the immune clonal selection algorithm to estimate the parameters of the Muskingum model, getting a higher precision than the other techniques. Mohon [8] proposed the genetic algorithm to estimate the parameters of two nonlinear Muskingum routing models. Chen and Yang [9] applied the Gray-encoded accelerating genetic algorithm in the

parameters optimization of the Muskingum model. Chu and Chang [10] used the Particle Swarm Optimization (PSO) algorithm to estimate the Muskingum model parameters, which improves the accuracy of the Muskingum model for flood routing. Barati [11] applied the parameter-setting-free technique, interfaced with a harmony search algorithm to the parameter estimation of the Muskingum model, and found good model parameter values. Zhang et al. [12] applied the Shuffle Complex Evolution Algorithm (SCE-UA) to optimize and estimate the discharge proportion coefficient  $x$  of the Muskingum equation and the number of streams subsections partitioned for mainstreams and tributaries. Niazkar and Afzali [13] proposed a hybrid method, which combined the Modified Honey Bee Mating Optimization and Generalized Reduced Gradient algorithms, reducing the sum of the squared (SSQ) value for the double-peak case study. Ouyang et al. [14] applied the invasive weed optimization (HIWO) algorithm to the parameter estimation of nonlinear Muskingum models. Although various techniques were applied to estimate the parameters of the nonlinear Muskingum flood routing model, the application of the model still suffers from a lack of an efficient method for parameter estimation. An efficient method for parameter estimation in the Muskingum model is still required to improve computational precision.

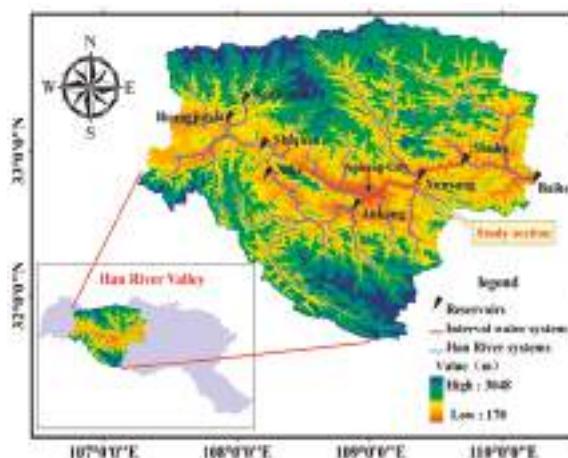
The improved Muskingum model mainly includes parameter setting and optimization criteria [15,16], and studies on model improvement have had a significant breakthrough in recent years. Zhang et al. and Vatankhah [17,18], for example, used a nonlinear Muskingum flood routing model with variable exponent parameters, producing the most accurate fit for outflow data. Moghaddam et al. [19] proposed a new Muskingum model with four parameters, the sum of the squared (SSQ) or absolute (SAD) deviations between the observed and estimated outflows considered as objective functions. Although the new Muskingum model becomes more complex, it improves the fit to observed flow, especially in multiple-peak hydrographs. Luo et al. [20] proposed a multi-objective estimation routine of the Muskingum model, involving single-peak, multi-peak, and non-smooth hydrographs, proving that the multi-objective estimation procedure is consistent and effective in estimating the parameters of the Muskingum model. Easa [21] pointed out models that adopt the outflow criterion result in a poor fit to the observed storages, presenting a new approach that incorporates both criteria in the estimation process and aids trade-off analysis.

This paper proposes a parameter estimation of the Muskingum model based on a new intelligent algorithm—the Wolf Pack Algorithm (WPA). In this paper, several floods with different magnitudes from river channels (Ankang hydropower station–Ankang city and Ankang hydropower station–Shuhe) of the Hanjiang River were first selected to estimate their respective parameters. Secondly, an improved multi-objective of the Muskingum model as well as that of the WPA and its solving steps are proposed. Thirdly, the results from the different algorithms in terms of the weighted sum of absolute difference and the coefficient of determination between the observed and routed outflows are compared to verify the performance of the WPA. Finally, conclusions are drawn based on the results.

## 2. Materials and Methods

### 2.1. Research Area

The Ankang to Shuhe section of Hanjiang River in China was chosen as the study area (Figure 1). The length of the river section is 109.36 km, accounting for 7% of the total reach of the Han River, with a drop of 61 m and an average gradient ratio of 0.06% of the river bed. Due to the steep slope of the mountain basin, the poor permeability of the rock formation, and poor adjustment of the channel, the flood process is very fast, the peak shape is thin, the inter-annual variability of the flood is larger, and the flood variation coefficient is larger. The parameters of the river and flood data are shown in Tables 1 and 2.



**Figure 1.** Map of the Hanjiang River and the location of major reservoirs.

**Table 1.** Data of river section.

Reach	Height Difference (m)	River Length (km)	Gradient (%)
Ankang Reservoir-Ankang city (Reach 1)	11.24	18.31	0.06
Ankang Reservoir-Shuhe Reservoir (Reach 2)	61.00	109.00	0.06

**Table 2.** Data of flood.

Reach	Flood Event	Flood Peak (m <sup>3</sup> /s)	Total Amount of Floods (10 <sup>8</sup> m <sup>3</sup> )	Flood Peak Time (h)
Ankang Reservoir– Ankang city	20100821	7260	173	1
	20120707	8420	241	1
	20120901	7645	171	1
Ankang Reservoir– Shuhe Reservoir	20130722	7178	44	7
	20140909	7607	67	6
	20140914	6637	52	7

## 2.2. Muskingum Model

### 2.2.1. Model Basic Principle

The Muskingum model is a traditional method for solving river flood routing, which is mainly solved by the continuity equation and the dynamic equation [22]. In order to obtain the flood routing equation, the water balance equation and storage equation are solved by:

$$Q_2 = C_0 I_2 + C_1 I_1 + C_2 Q_1 \quad (1)$$

$$\left\{ \begin{array}{l} C_0 = \frac{-Kx+0.5\Delta t}{K-Kx+0.5\Delta t} \\ C_1 = \frac{Kx+0.5\Delta t}{K-Kx+0.5\Delta t} \\ C_2 = \frac{K-Kx-0.5\Delta t}{K-Kx+0.5\Delta t} \end{array} \right. \quad (2)$$

where  $Q_1$  and  $Q_2$  are the outflow of the downstream section at the beginning and the end of the period, respectively,  $I_1$  and  $I_2$  are the inflow of upstream section at the beginning and the end of the period,

$K$  is channel storage coefficient, and  $x$  is specific gravity coefficient of flow.  $C_0$ ,  $C_1$ , and  $C_2$  are the flow routing coefficients.

### 2.2.2. Design of Objective Function

The optional parameters of the Muskingum model can be  $x$  and  $K$ , or  $C_0$ ,  $C_1$ , and  $C_2$ . If  $K$  and  $x$  are chosen as the optimization variables, it is necessary to ensure that the water storage capacity of the river is a single linear relationship with the reservoir flow during the optimization process, which is difficult to determine, given the range of variables. Meanwhile, there is no need to go deep into the physical parameters of the model, because of the intelligent algorithm based on the black box model [23]. Therefore,  $C_0$  and  $C_1$  are chosen as the optimization variables. The results of river flood routing are mainly reflected in the flood process and the flood peak of the simulation.

The results of river flood flow evolution are mainly reflected in the degree of fitting of flood process and flood peak to the actual flood. Some studies indicated that the success of a calibration process is highly dependent on the objective function chosen as a calibration criterion [19]. The most commonly used objective function for the calibration procedure is the SSQ errors between observed and computed outflow [4,5,20,24], but some research has indicated that the SSQ is not necessarily correct [16,25]. Considering the above reasons, the objective function established in this paper can be described as follows:

Objective 1: Minimum weighted sum of absolute difference (SAD):

$$f_1 = \min \sum_{i=2}^n |Q_0(i)|C_0 I_c(i) + C_1 I_c(i-1) + C_2 Q_0(i-1) - Q_0(i)| \quad (3)$$

$$\begin{aligned} \text{s.t. :} \quad & C_0 \in [-1, 1] \\ & C_1 \in [-1, 1] \\ & C_2 = 1 - C_0 - C_1 \in [-1, 1] \end{aligned} \quad (4)$$

where  $Q_0(i)$  is the actual outflow of the downstream section at a time  $i$ ,  $I_c(i-1)$  and  $I_c(i)$  are the simulated flow for the upstream section at time  $(i-1)$  and  $i$ , respectively,  $Q_0(i-1)$  is the simulated flow for the upstream section at time  $(i-1)$ , and  $n$  is the length of the time series used for calibration.  $C_0$ ,  $C_1$ , and  $C_2$  are the flow routing coefficients.

The SAD will give the minimum difference between observed and computed outflow [9,19,26]. Objective 1 is the SAD multiplying the corresponding weight taken from the observed flow at the corresponding time, which will increase large flow influence on the parameter estimation, especially on the flood peak. Thus, the weight can increase the simulation accuracy of the flood peak.

Objective 2: Maximum coefficient of determination  $D_c$ :

$$f_2 = \max D_c = \max \left[ 1 - \frac{\sum_{i=1}^n (Q_c(i) - Q_0(i))^2}{\sum_{i=1}^n (Q_0(i) - \bar{Q}_0)^2} \right] \quad (5)$$

where  $D_c$  is the coefficient of determination,  $Q_c(i)$  is the computed value at time  $i$ ,  $Q_0(i)$  is the observed value at time  $i$ ,  $\bar{Q}_0$  is the mean value of the observed flood process, and  $n$  is the number of time series used for calibration [22].

The deterministic coefficient is an indicator to measure the consistency between the flood forecast and the observed process. In this paper, the coefficient of determination as Objective 2 can ensure that the simulated flow process is the closest to the observed values [27,28].

In order to reduce the multi-objective calculation and adaptation algorithm, the comprehensive multi-objective function chosen in this paper is as follows:

$$f = \max(\phi_1(-f_1) + \phi_2 f_2) \quad (6)$$

where  $f$  is the comprehensive objective,  $f_1$  and  $f_2$  are Objective 1 and Objective 2, respectively, and  $\phi_1$  and  $\phi_2$  are the weight coefficient of different objectives, respectively.

### 3. Methodology

#### 3.1. Wolf Pack Algorithm (WPA)

The Wolf Pack Algorithm is a novel swarm intelligence algorithm with strong local search ability and global convergence. It consists of the leading wolf, fierce wolves, and explore wolves, including three kinds of intelligent behaviors: Scouting, summoning, and beleaguered. Simultaneously, with a productive rule for the leading wolf, which is that the winner can dominate all, it is a renewable mechanism, namely survival of the fittest, for a pack of wolves. It has been applied in the field of mathematics, physics, and hydropower station optimization, achieving good calculation results [29–31]. In this paper, the WPA is used for the parameter estimation of the Muskingum model. Combining the principle of the WPA with the objective function, we designed a WPA for solving Muskingum model parameters. The procedure of our algorithm for parameter optimal estimation of the Muskingum model is shown as follows:

Step 1: (Initial wolves) We must initially determine the size of the algorithm population  $n$  and the number of iterations  $gen$ , the probe wolf scale factor  $a$ , the population regeneration factor  $b$ , the step factor  $S$ , the distance determination factor  $W$ , and the maximum number of scouting  $T_{\max}$ . The Model Parameter ( $C_0, C_1$ ) is regarded as the position of the artificial wolf ( $x_1, x_2$ ) in a two-dimensional decision space. The position of the wolf pack is initialized in the range of  $C_0$  and  $C_1$ , as shown in Equation (7):

$$x_{i,j} = 2 \times \text{rand} - 1 \quad i = 1, 2, \dots, n; j = 1, 2 \quad (7)$$

where  $x_{i,j}$  is the initial position of the  $i$ -th wolf in the  $j$ -th decision space;  $\text{rand}$  is a uniformly distributed random number in the interval [0,1].

Step 2: (Scouting) Choose the objective function  $f$  as the prey odor concentration  $Y_t(i)$  and calculate the prey odor concentration at the location of the artificial wolf. According to the  $Y_t(i)$ , sort all artificial wolf positions in descending order, then select the first artificial wolf of the sorted population as a head wolf, whose location is regarded as  $x_{lead}$  and the prey odor concentration is  $Y_{lead}$ . Choose the second to  $S + 1$  artificial wolves as the explore wolves that total  $S = \text{Round}(a * n)$ , and the remaining artificial wolves as the fierce wolves. The explore wolves scout to conduct a fine search, until the maximum number of walks is reached or the maximum global optimal solution is found.

Step 3: (Summoning) The fierce wolfs quickly approach the head wolf to achieve the global convergence of solution, until the Euclidean distance  $d_{is}(i)$  between all fierce wolves and the head wolf is less than the judgment value of distance  $d_{near}$  or a contemporary global optimal solution is found. The calculation formula of the distance judgment value is:

$$d_{near} = \frac{\sum_{d=1}^D |\max^d - \min^d|}{D * W} = \frac{2}{W} \quad (8)$$

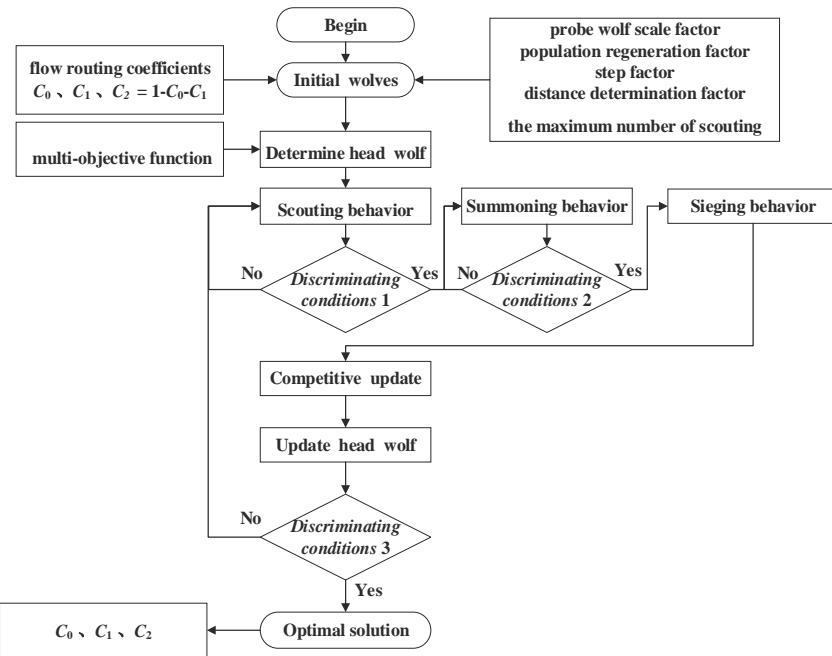
where  $d_{near}$  is the judgment value of distance;  $D$  is the spatial dimension;  $W$  is the distance determining factor;  $\max^d, \min^d$  are the upper and lower limits of the decision variable  $x_{i,j}$ , respectively.

Step 4: (Sieging) Under the command of the head wolf, other artificial wolves further update the location to achieve local fine optimization.

Step 5: (Competitive update) According to the population, update factor  $b$ , calculate the amount of the population, update  $R$  and regenerate  $R$  artificial wolves, which can be alternated with the previous generation of artificial wolves with a low fitness value to ensure the diversity of solutions and to avoid falling into the local optimal.

Step 6: Determine whether the maximum number of iterations has been reached; if it has, output the optimal global solution. If not, proceed to the next generation calculation and return to Step 2 until the maximum number of iterations is reached.

The WPA flow chart for solving the Muskingum model parameters is shown in Figure 2.



**Figure 2.** Flow chart of parameter estimation.

### 3.2. Parameter Sensitivity Analysis

The WPA involves a relatively large number of parameters; the main sensitivity parameters are the distance determination factor  $W$  and the step factor  $S$ . Therefore, this paper focuses on the discussion of parameters  $W$  and  $S$  on the algorithm. In order to determine the distance determination factor  $W$  and the step factor  $S$ , the parameters were set as follow: Artificial wolf population size  $n = 20$ , explore wolf proportion factor  $a = 4$ , population update factor  $b = 2$ , the maximum number of scouting  $T_{\max} = 2$ , the number of iterations  $gen = 30$ . With the Muskingum principle, the algorithm is used to solve  $D$ -dimensional space problem ( $D = 2$ ), and the upper and lower limits of the variables are  $\max^d = 1$  and  $\min^d = -1$ , respectively.

The WPA's raid step:

$$step_b^d = \frac{2(\max^d - \min^d)}{S} = \frac{4}{S} \quad (9)$$

where  $step_b^d$  is the step size of the  $d$ -dimensional space;  $S$  is the step factor;  $\max^d, \min^d$  are the upper and lower limits of the decision variable  $x_{i,j}$ , respectively.

In the summoning act, the condition of the artificial wolf raid termination is  $d_{is} < d_{near}$ , so the rapid step should meet the following formula:

$$\sum_{d=1}^D step_b^d < d_{near} \quad (10)$$

where the parameters are the same as above.

From Formula (10):

$$S > 4W \quad (11)$$

where the parameters are the same as above.

From the above derivation, it can be seen that the normal operation of the algorithm can be satisfied when the step factor  $S$  and the distance determination factor  $W$  satisfy the formula (11).

Fixed distance determination factor  $W = 20$ , set  $S = 100, 105, 110, 115, 120$ , and  $130$ , respectively. For each  $S$ , run the program code 20 times independently, select the evaluation index of the algorithm, the absolute deviation of the flood process, and the observed and simulated flood peak deviation. The absolute deviation of the flood process and the formula are as follow:

$$\delta = \sum_{i=1}^n |(Q_0(i) - Q_c(i))| \quad (12)$$

where  $\delta$  is the absolute deviation of the flood process,  $\text{m}^3/\text{s}$ .

The flood peak deviation is:

$$f = \frac{1}{n} \sum_{i=1}^n |(Q_{obs,i}^{peak} - Q_{sim,i}^{peak})| \quad (13)$$

where  $f$  is the flood peak deviation;  $Q_{obs,i}^{peak}$  and  $Q_{sim,i}^{peak}$  are the observed and simulated maximum outflow at peak flow event number  $i$ , respectively;  $n$  is the number of simulations times.

The evaluation index values are taken from the average of 20 independent runs. Taking the flood event of 20100821 as an example, the results are shown in Table 3.

**Table 3.** Sensitivity analysis of step factor.

<b>W</b>	<b>S</b>	<b><math>\delta</math> (<math>\text{m}^3/\text{s}</math>)</b>	<b>Flood Peak Deviation (<math>\text{m}^3/\text{s}</math>)</b>
20	100	7892.7	6.33
	104	7892.4	6.31
	110	7891.9	6.29
	115	7891.7	6.29
	120	7891.6	6.29
	130	7891.6	6.29

As shown in Table 3:

(1) With the increase of the  $S$  value,  $\delta$  decreases slightly from 7892.7  $\text{m}^3/\text{s}$  to 7891.6  $\text{m}^3/\text{s}$ , which indicates that the larger the step size, the finer the search, and the closer the flood forecast results are to the actual flood process.

(2) When  $W = 20$ , the variation range of  $\delta$  and flood peak are not large, which indicates that the step size of the WPA has little influence on the forecast results in a certain range.

Keeping in mind that the smaller the searching step, the more time consuming the process is, the initial selection of  $S = 120$ , that is  $S = 6W$ .  $W = 20, 30, 40, 50, 60$ , and  $70$  are selected to simulate the same flood, and the results are shown in Table 4.

As shown in Table 4:

(1) With the increase of  $W$ ,  $\delta$  decreases slightly, and the minimum value appears at  $W = 60$ . The peak deviation is also minimized, and then the  $W$  value increases; the optimization effect has no obvious change.

(2) The range of  $\delta$  is not significant, and indicates that the parameter  $W$  of the wolves algorithm on the two-dimensional nonlinear optimization problem has little influence on the algorithm, and the algorithm is more applicable.

**Table 4.** Sensitivity analysis of distance judgment factor.

<i>W</i>	<i>S</i>	$\delta$ ( $m^3/s$ )	Flood Peak Deviation ( $m^3/s$ )
20	120	7891.6	6.29
30	180	7891.4	6.20
40	240	7891.2	6.20
50	300	7891.0	6.12
60	360	7890.8	6.12
70	420	7890.8	6.12

With the increase of *W*, the raid step will be smaller, and the optimization result is too fine, which will cause the artificial wolf to turn from difficult to besieging. Thus, the algorithm has the possibility of entering an infinite loop.

In order to further analyze the influence of different weight coefficients on the forecast results, we set different weight coefficients using the same method. The results are shown in Table 5.

**Table 5.** Multi-objective weight coefficient analysis.

$\phi_1$	$\phi_2$	$\delta$ ( $m^3/s$ )	Flood Peak Deviation ( $m^3/s$ )
0.9	0.1	7916.9	6.16
0.8	0.2	7914.8	6.16
0.7	0.3	7912.9	6.18
0.6	0.4	7906.1	6.19
0.5	0.5	7901.0	6.19
0.4	0.6	7896.6	6.19
0.3	0.7	7891.1	6.20
0.2	0.8	7891.7	6.22
0.1	0.9	7891.6	6.26

As shown in Table 5:

- (1) With the decrease of  $\phi_1$ , peak deviation increased slightly, with an increase 0.1  $m^3/s$  only. When  $\phi_1$  is 0.8 or 0.9, the flood peak deviation takes the minimum, which indicated that the weight factor of Objective 1 focuses on the minimum deviation of flood peak.
- (2) With the increase of  $\phi_2$ , the absolute deviation of the flood process is gradually reduced from 7916.9  $m^3/s$  to 7891.6  $m^3/s$ , and reaches the minimum value when  $\phi_2$  is 0.7, which indicated that the weight factor of Objective 2 focuses on the total error of the entire flood process.

According to comprehensive analysis, the weight coefficient  $\phi_2$  had a great influence on the fitting effect of the whole flood process and  $\phi_1$  had little effect on the flood peak simulation. Therefore,  $\phi_1 = 0.3$  and  $\phi_2 = 0.7$  were selected for the multi-objective weight value.

#### 4. Results and Discussion

In this paper, we took flood forecasting of Ankang hydropower station to Ankang city and Ankang hydropower station to Shuhe hydropower Station as examples. Considering the length of the two Reaches, the actual demand, and the characteristics of the inflow, the calculation time steps of Reach 1 and Reach 2 were 1 h and 6 h, respectively. The improved Muskingum method (WPA), Particle Swarm Optimization (PSO), and Trial Algorithm (TA) were selected to compare to the observed outflow. In the WPA, the values of distance determination (*W*) and the step (*S*) were both 100. The maximum number of iterations was 50. The value of the artificial wolf population size (*n*) was 20. The values of the explore wolf proportion (*a*) and population update (*b*) were 4 and 2, respectively. In the TA [22], the parameters (*k*, *x*) of this algorithm are shown in Table 6. In the PSO algorithm, a population of 50 individuals was used. The maximum number of iterations in the program was 100. The values of the acceleration constants  $c_1$  and  $c_2$  were both set to 0.2.

The corresponding parameters estimated by the WPA, TA, and PSO are listed in Table 6. It can be seen from Table 6 that the parameters obtained from the three methods are all technically reasonable, but do show marked differences. The estimated value of the parameters differ very little between the WPA and PSO, but the parameters of TA vary quite significantly with WPS and PSO.

**Table 6.** Parameter estimation results obtained from different methods.

Flood Event	20100821	20120707	20120901	20130722	20140909	20140914
TA	$k$	1.50	1.50	1.00	1.00	1.00
	$x$	0.10	0.10	0.20	0.20	0.20
PSO	$k$	0.21	1.18	0.88	1.48	1.37
	$x$	-4.17	0.23	-0.43	0.23	-0.42
WPA	$k$	0.58	1.23	0.86	1.46	1.56
	$x$	-0.88	0.24	-0.42	0.22	-0.68

The comparison of the observed and simulation results of these three methods is presented in Figures 3 and 4.

As shown in Figures 3 and 4:

(1) No matter the length of the river, the simulation results of the WPA and PSO compared to those of the TA have a high fitting effect. It is embodied in the following aspects: The simulated peak values are the same as the observed ones, and the simulation process of the backwater segment is almost coincident with the observed process. However, the simulated values by the WPA and PSO in the rising water segment are different from the observed ones. Compared with the observed values, the deviation errors of the simulation values of the WPA are smaller than those of the PSO, and the deviation errors of the simulation values of the TA are the biggest.

(2) The fitting degree of the flood process in short reach is higher than that of the long river section by the WPA, PSO, and TA. Regardless of the length of the river, the error of the TA is the largest, and the WPA simulation deviation is slightly lower than that of the PSO.

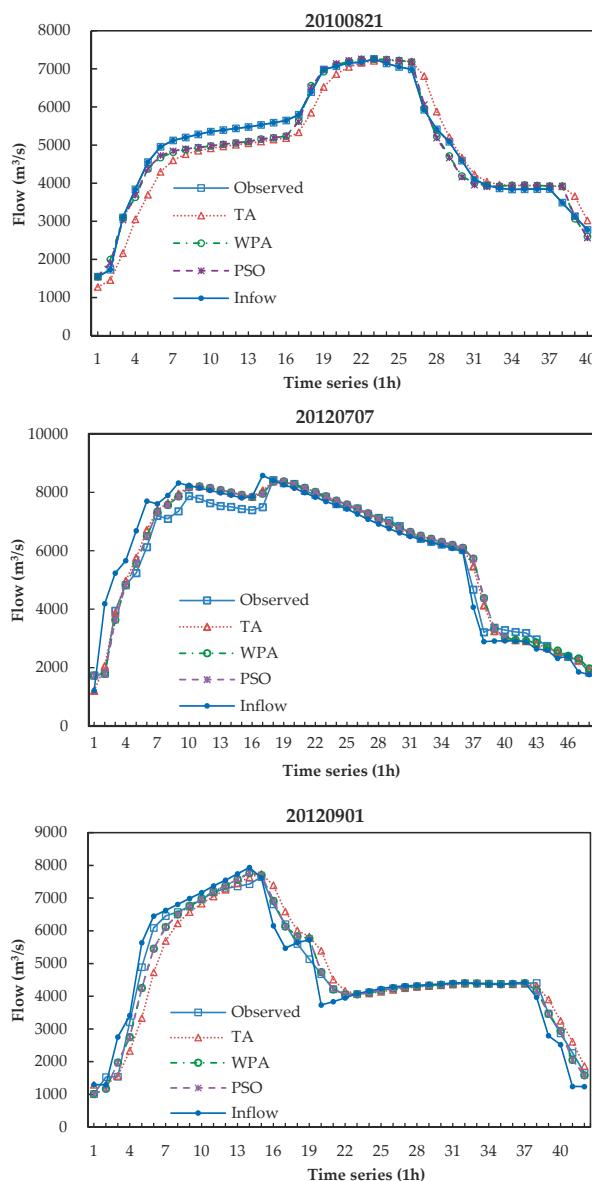
In order to visualize the fitting effect of the three methods, the 20140909 flood of Ankang Power Station to Shuhe Power Station was selected as an example (Table 7), and the correlation diagram of observed and simulated flow are drawn in Figure 5.

As shown in Figure 5:

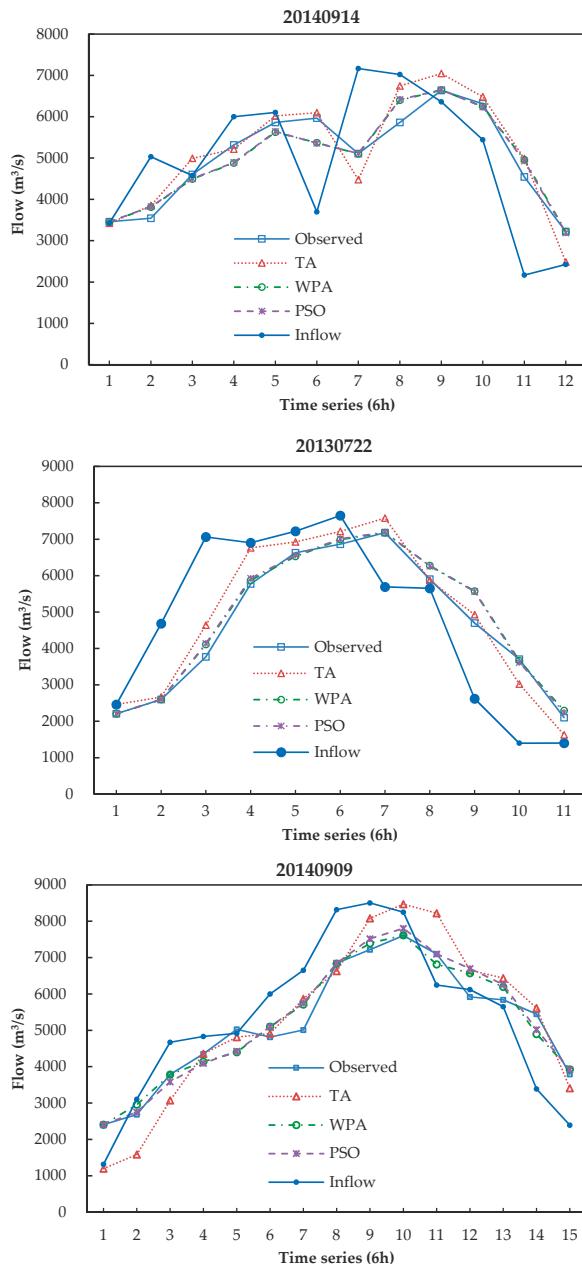
(1) The flooding process of the WPA simulation has 4 points falling on the line, and the trend line coefficient of WPA simulated flood process is 0.99, which is close to 1 compared to that of the PSO and TA, which indicates that the WPA has a high precision in the simulation of the whole flood process.

(2) The simulated flood peaks of the WPA fall in the straight line  $y = x$ , which indicates that the simulation value is equal to the observed. The peak values of the PSO and TA fall on the upper side of the line  $y = x$ , which indicates that the WPA has better simulation effect on peak value.

In order to further assess the precision of the WPA on the flood process and flood peak transmission time, the absolute deviation ( $\delta$ ) of flood process, flood peak deviation, and flood peak transmission time error are selected as the evaluation indices. Tables 8 and 9 show the indices values by the WPA, PAO, and TA of different floods.



**Figure 3.** Simulation results of different floods in Reach 1.



**Figure 4.** Simulation results of different floods in Reach 2.

As shown in Tables 8 and 9:

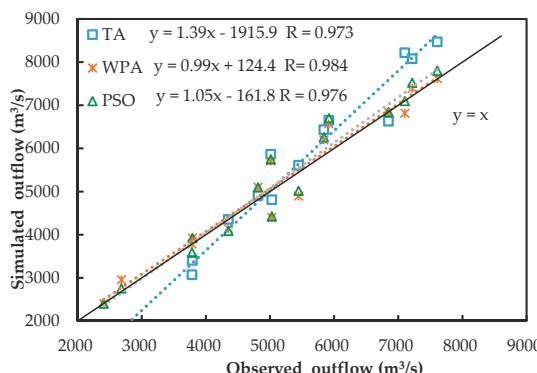
(1) The WPA and PSO simulation results have a significant advantage over the results of the TA. Compared to that of the TA, the depreciation of absolute deviation cumulative value of flood process calculated by the WPA and PSO are 53% and 52%, respectively. The depreciation of absolute deviation

of flood peak calculated by the WPA and PSO are 99% and 77% based on the 20140909 flood of Ankang power station–Shuhe hydropower station, respectively.

(2) The maximum deviations of the WPA, PSO, and TA are  $9.5 \text{ m}^3/\text{s}$ ,  $197.1 \text{ m}^3/\text{s}$  and  $865.8 \text{ m}^3/\text{s}$ , respectively. Moreover, the minimum deviations are  $0.3 \text{ m}^3/\text{s}$ ,  $15 \text{ m}^3/\text{s}$ , and  $402 \text{ m}^3/\text{s}$ , respectively, which indicate that the flood peak simulated by the WPA and PSO has obvious advantages over that of the TA. In the 20140909 flood simulation, flood peak deviations are  $9.5 \text{ m}^3/\text{s}$  by the WPA and  $197.1 \text{ m}^3/\text{s}$  by the PSO, respectively. Flood peak deviation simulated by the WPA is much smaller than that of the PSO, which indicates that the WPA can significantly improve peak forecast accuracy.

**Table 7.** 20140909 flood simulation of Ankang Power Station to Shuhe Power Station.

Time	Inflow ( $\text{m}^3/\text{s}$ )	Outflow ( $\text{m}^3/\text{s}$ )				Relative Error (100%)		
		Observed	TA	WPA	PSO	TA	WPA	PSO
0	1315	2405.47	1196.50	2405.47	2405.47	50.26%	0.00%	0.00%
6	3105	2682.43	1585.65	2956.10	2682.43	40.89%	10.20%	2.73%
12	4671	3783.79	3076.53	3784.10	3783.79	18.69%	0.01%	5.22%
18	4830	4351.65	4358.84	4141.88	4351.65	0.17%	4.82%	5.96%
24	4916	5026.62	4813.45	4400.83	5026.62	4.24%	12.45%	11.93%
30	5998	4813.56	4914.02	5102.37	4813.56	2.09%	6.00%	5.96%
36	6652	5010.76	5867.91	5709.84	5010.76	17.11%	13.95%	14.74%
42	8318	6846.56	6631.47	6836.78	6846.56	3.14%	0.14%	0.00%
48	8505	7217.19	8082.39	7395.13	7217.19	11.99%	2.47%	4.20%
54	8251	7607.34	8473.15	7616.87	7607.34	11.38%	0.13%	2.59%
60	6247	7101.53	8221.21	6816.21	7101.53	15.77%	4.02%	0.01%
66	6120	5917.64	6654.63	6574.14	5917.64	12.45%	11.09%	13.25%
72	5649	5839.00	6438.00	6196.58	5839.00	10.26%	6.12%	7.21%
78	3390	5446.03	5613.68	4897.68	5446.03	3.08%	10.07%	7.84%
84	2390	3789.75	3409.38	3925.44	3789.75	10.04%	3.58%	3.53%



**Figure 5.** Correlation of observed and simulation data.

**Table 8.** Analysis of Ankang hydropower station–Ankang city simulation results.

Floods	20100821				20120707				20120901			
	Method	TA	PSO	WPA	TA	PSO	WPA	TA	PSO	WPA	TA	PSO
$\delta (\text{m}^3/\text{s})$		13,960	7892	7865	11,404	10,606	10,607	197,269	5529	5528		
Flood peak deviation ( $\text{m}^3/\text{s}$ )		34.3	6.5	0.5	33	51.6	51.4	87.76	52.1	51.7		
Flood peak time transmission error (h)		1	0	0	0	0	0	1	0	0		

**Table 9.** Analysis of Ankang hydropower station–Shuhe hydropower station simulation results.

Floods	20130722			20140909			20140914		
Method	TA	PSO	WPA	TA	PSO	WPA	TA	PSO	WPA
$\delta$ ( $m^3/s$ )	4666	2221	2162	9141	4422	4278	4389	2697	2678
Flood peak deviation ( $m^3/s$ )	402.0	19.5	0.3	865.8	197.1	9.5	409.0	15.0	8.6
Flood peak time transmission error (h)	0	0	0	0	0	0	0	0	0

## 5. Conclusions

In this paper, a new multi-objective Muskingum model was established and solved by a new algorithm called the WPA, and its performance was verified by using the Ankang to Shuhe section of Hanjiang River datasets. The following work has been done:

(1) After a brief literature review, a novel parameter estimation approach based on the WPA was proposed for parameter estimation of the nonlinear Muskingum flood, which considered two calibration objectives in the calibration procedure: (1) The weighted sum of absolute difference between the routed and observed outflows, and (2) the coefficient of determination between the routed and observed outflows.

(2) The proposed approach is compared to the other methods (TA, PSO) for an example case from the Hanjiang River, and the results demonstrate that the WPA can achieve a high degree of accuracy to estimate the parameters and results in accurate predictions of outflow.

(3) In this study, however, the parameters of WPA, such as the distance judgment factor and the step factor, have limited its application range. A possible reason is that different rivers have its applicable parameters. In future research, more floods in different rivers basins will be used to expand its range of applications.

**Author Contributions:** T.B. and Q.H. provided the ideas and supervised the study; W.Y. conceived and designed the methods; T.B. and J.W. wrote the paper, and all the authors were responsible for data processing and data analysis.

**Funding:** This study is supported by the National Key R&D Program of China (2017YFC0405900); Planning Project of Science and Technology of Water Resources of Shaanxi (2017slkj-16, 2017slkj-27); the China Postdoctoral Science Foundation (2017M623332XB); and the Basic Research Plan of Natural Science in Shaanxi Province (2018JQ5145).

**Acknowledgments:** The authors are grateful to the three reviewers who helped us to sustain high quality papers in the review process. The above financial support is gratefully acknowledged.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Kuriqi, A.; Ardiçlioğlu, M.; Muceku, Y. Investigation of Seepage Effect on River Dike's Stability under Steady State and Transient Conditions. *Pollack Period.* **2016**, *11*, 87–104. [[CrossRef](#)]
- Sajikumar, N.; Gyncy, I.; Sumam, K.S. Modelling of Nonlinear Muskingum Method using Control System Concept. *Aquat. Procedia* **2015**, *4*, 979–985. [[CrossRef](#)]
- Mccarthy, G. The unit hydrograph and flood routing. In Proceedings of the Conference of the North Atlantic Division, U.S. Engineer Department, New London, CT, USA, 24 June 1938; pp. 608–609.
- Luo, J.; Xie, J. Discussion of “Parameter Estimation for Nonlinear Muskingum Model Based on Immune Clonal Selection Algorithm”. *J. Hydrol. Eng.* **2010**, *15*, 391–392. [[CrossRef](#)]
- Xu, D.M.; Qiu, L.; Chen, S.Y. Estimation of Nonlinear Muskingum Model Parameter Using Differential Evolution. *J. Hydrol. Eng.* **2012**, *17*, 348–353. [[CrossRef](#)]
- Gill, M.A. Flood routing by the Muskingum method. *J. Hydrol.* **1978**, *36*, 353–363. [[CrossRef](#)]
- Kuriqi, A.; Ardiçlioğlu, M. Investigation of hydraulic regime at middle part of the Loire River in context of floods and low flow events. *Pollack Period.* **2018**, *13*, 145–156. [[CrossRef](#)]
- Mohon, S. Parameter Estimation of Nonlinear Muskingum Models Using Genetic Algorithm. *J. Hydraul. Eng.* **1997**, *123*, 137–142. [[CrossRef](#)]

9. Chen, J.; Yang, X. Optimal parameter estimation for Muskingum model based on Gray-encoded accelerating genetic algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2007**, *12*, 849–858. [[CrossRef](#)]
10. Chu, H.J.; Chang, L.C. Applying Particle Swarm Optimization to Parameter Estimation of the Nonlinear Muskingum Model. *J. Hydrol. Eng.* **2009**, *14*, 1024–1027. [[CrossRef](#)]
11. Barati, R. Discussion of Parameter Estimation of the Nonlinear Muskingum Model Using Parameter-Setting-Free Harmony Search. *J. Hydrol. Eng.* **2011**, *17*, 1414–1416. [[CrossRef](#)]
12. Zhang, H.; Li, Z.; Kan, G.; Yao, C. Muskingum parameter estimation method based on SCE-UA algorithm and its realization in programming. *J. Hydraul. Eng.* **2013**, *32*, 43–47. (In Chinese)
13. Niazkar, M.; Afzali, S.H. Application of New Hybrid Optimization Technique for Parameter Estimation of New Improved Version of Muskingum Model. *Water Resour. Manag.* **2016**, *30*, 4713–4730. [[CrossRef](#)]
14. Ouyang, A.; Liu, L.B.; Sheng, Z.; Wu, F. A Class of Parameter Estimation Methods for Nonlinear Muskingum Model Using Hybrid Invasive Weed Optimization Algorithm. *Math. Probl. Eng.* **2015**, *2015*, 573894. [[CrossRef](#)]
15. Gupta, H.V.; Sorooshian, S.; Yapo, P.O. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resour. Res.* **1998**, *34*, 3663–3674. [[CrossRef](#)]
16. Bekele, E.G.; Nicklow, J.W. Multi-objective automatic calibration of SWAT using NSGA-II. *J. Hydrol.* **2007**, *341*, 165–176. [[CrossRef](#)]
17. Zhang, S.; Kang, L.; Zhou, L.; Guo, X. A new modified nonlinear Muskingum model and its parameter estimation using the adaptive genetic algorithm. *Hydrol. Res.* **2017**, *48*, 17–27. [[CrossRef](#)]
18. Vatankhah, A.R. Nonlinear Muskingum model with inflow-based exponent. *Water Manag.* **2016**, *170*, 66–80. [[CrossRef](#)]
19. Moghaddam, A.; Behmanesh, J.; Farsijani, A. Parameters Estimation for the New Four-Parameter Nonlinear Muskingum Model Using the Particle Swarm Optimization. *Water Resour. Manag.* **2016**, *30*, 2143–2160. [[CrossRef](#)]
20. Luo, J.; Zhang, X.; Zhang, X. Multi-Objective Calibration of Nonlinear Muskingum Model Using Non-Dominated Sorting Genetic Algorithm-II. *Int. Conf. Appl. Math. Simul. Model.* **2016**. [[CrossRef](#)]
21. Easa, S.M. Multi-criteria optimization of the Muskingum flood model: A new approach. *Water Manag.* **2015**, *168*, 220–231. [[CrossRef](#)]
22. Bao, W.M.; Zhang, J.Y. *Hydrologic Forecasting*, 4th ed.; China Water and Power Press: Beijing, China, 2009; pp. 3–93. (In Chinese)
23. Liu, C.Y.; Yan, X.H.; Wu, H. The Wolf Colony Algorithm and Its Application. *Chin. J. Electron.* **2011**, *20*, 212–216. (In Chinese)
24. Karahan, H.; Gurarslan, G.; Zong, W.G. Parameter Estimation of the Nonlinear Muskingum Flood Routing Model Using a Hybrid Harmony Search Algorithm. *J. Hydrol. Eng.* **2013**, *18*, 352–360. [[CrossRef](#)]
25. Easa, S.M. New and improved four-parameter non-linear Muskingum model. *Water Manag.* **2015**, *167*, 288–298. [[CrossRef](#)]
26. Zhang, G.; Xie, T.; Zhang, L.; Hua, X.; Wu, C.; Chen, X.; Li, F.; Zhao, B. “In-Process Type” Dynamic Muskingum Model Parameter Estimation Method. *Water* **2017**, *9*, 849. [[CrossRef](#)]
27. Song, X.M.; Kong, F.Z.; Zhu, Z.X. Application of Muskingum routing method with variable parameters in ungauged basin. *Water Sci. Eng.* **2011**, *4*, 1–12. [[CrossRef](#)]
28. Franchini, M.; Bernini, A.; Barbetta, S.; Moramarco, T. Forecasting discharges at the downstream end of a river reach through two simple Muskingum based procedures. *J. Hydrol.* **2011**, *399*, 335–352. [[CrossRef](#)]
29. Wu, H.S.; Zhang, F.M.; Wu, L. New swarm intelligence algorithm-wolf pack algorithm. *Syst. Eng. Electron.* **2013**, *35*, 2430–2438. [[CrossRef](#)]
30. Lai, L.H. Research on gyroscope random drift modeling based on ARMA model and wolf pack algorithm. *Transducer Microsyst. Technol.* **2016**, *35*, 56–58. (In Chinese)
31. Wang, J.; Jiao, Y. Improvement of wolf pack search algorithm and its application to optimal operation of reservoirs. *Eng. J. Wuhan Univ.* **2017**, *50*, 161–167. (In Chinese)



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Flash-Flood Forecasting in an Andean Mountain Catchment—Development of a Step-Wise Methodology Based on the Random Forest Algorithm

Paul Muñoz <sup>1,2,\*</sup>, Johanna Orellana-Alvear <sup>1,3</sup>, Patrick Willems <sup>2</sup> and Rolando Céller <sup>1,4</sup>

<sup>1</sup> Departamento de Recursos Hídricos y Ciencias Ambientales, Universidad de Cuenca, Cuenca 010150, Ecuador; johanna.orellanaa@gmail.com (J.O.-A.); rolando.celleri@ucuenca.edu.ec (R.C.)

<sup>2</sup> Department of Civil Engineering, Hydraulics Section, KU Leuven, 3001 Leuven, Belgium; patrick.willems@kuleuven.be

<sup>3</sup> Laboratory for Climatology and Remote Sensing, Faculty of Geography, University of Marburg, 35032 Marburg, Germany

<sup>4</sup> Facultad de Ingeniería, Universidad de Cuenca, Cuenca 010150, Ecuador

\* Correspondence: paul.andres.munoz@gmail.com; Tel.: +593-995-981-786

Received: 31 August 2018; Accepted: 12 October 2018; Published: 26 October 2018

**Abstract:** Flash-flood forecasting has emerged worldwide due to the catastrophic socio-economic impacts this hazard might cause and the expected increase of its frequency in the future. In mountain catchments, precipitation-runoff forecasts are limited by the intrinsic complexity of the processes involved, particularly its high rainfall variability. While process-based models are hard to implement, there is a potential to use the random forest algorithm due to its simplicity, robustness and capacity to deal with complex data structures. Here a step-wise methodology is proposed to derive parsimonious models accounting for both hydrological functioning of the catchment (e.g., input data, representation of antecedent moisture conditions) and random forest procedures (e.g., sensitivity analyses, dimension reduction, optimal input composition). The methodology was applied to develop short-term prediction models of varying time duration (4, 8, 12, 18 and 24 h) for a catchment representative of the Ecuadorian Andes. Results show that the derived parsimonious models can reach validation efficiencies (Nash-Sutcliffe coefficient) from 0.761 (4-h) to 0.384 (24-h) for optimal inputs composed only by features accounting for 80% of the model's outcome variance. Improvement in the prediction of extreme peak flows was demonstrated (extreme value analysis) by including precipitation information in contrast to the use of pure autoregressive models.

**Keywords:** flash-flood; precipitation-runoff; forecasting; lag analysis; random forest; machine learning

---

## 1. Introduction

Flooding is nowadays the most common natural disaster worldwide [1]. Its occurrence is expected to further intensify during coming years due to the increase in the frequency of extreme precipitation events [2]. As a result, during last decades, real-time flood forecasting has become an emerging field of research for water management and risk analysis [3]. In highland catchments, extreme flash flood events have the potential to cause serious damage to downstream infrastructure and produce large socio-economic impacts. In the Andean region of Ecuador, flash-flood events cause human losses and perturb the everyday life of people (e.g., interruption in the water supply service, damage in transportation network) [4]. According to a report of the Andean community for the period 1970–2007 (see <http://www.comunidadandina.org>), 263 flash-floods events and 357 landslides (as a side effect, mostly reported in the city of Cuenca) have caused 429 human deaths as well as partial and complete destruction of 1568 and 581 houses, respectively.

The flash-flood hydrological response is highly non-linear since stream flow processes are complex phenomena exhibiting high spatial and temporal variability [5–8]. The main flash-flood driving forces are spatial and temporal precipitation variability, topography and soil humidity characteristics [8,9]. Nevertheless, in the Andean region, flash-flood forecasting is challenging considering that information other than precipitation and discharge is not commonly available due to budget constraints, the remoteness of the study areas and more importantly due to the extreme variability of the main driving forces previously mentioned. As a result, a simple, yet useful, approach is the development of precipitation-runoff forecasting models. Regional and local topography together with climatic influences are responsible for the spatio-temporal variation of precipitation, which is more prominent in mountainous areas. Specifically, in the Andean region, precipitation experiences extreme variability [10–12], and its characteristics are different for the eastern and western slopes of the Andean cordillera [13,14].

For predictive modeling, the use of machine learning (ML) techniques has recently and remarkably increased [15]. ML models rely on data-driven black-box processes aimed to infer from observations the stochastic dependency between the past and future [15]. These models are characterized by a more compact representation and high predictive potential, with considerably fewer parameters and variables when compared to fully distributed models [7].

Several ML methods have been developed: artificial neural networks (ANNs), support vector machines (SVMs), and decision trees (DTs), among others. Nevertheless, these methods exhibit some weaknesses such as overfitting for ANNs [16], the complexity of mathematical functions for SVMs [17] and the considerable effort needed to pre-process data for DTs [18]. In addition, unlike DTs, ANNs and SVMs are not able to estimate the relative importance of the features used in the model's input [19]. Random forest (RF) is a multiple DT-based algorithm (see [20]) characterized for its high predictive accuracy and its ability to perform a feature sensitivity analysis [21–23]. The RF algorithm can be used for classification and regression applications. However, although promising, RF (specially for regression) has been rarely used and evaluated in water resources studies [22].

For classification purposes, the study of Wang et al. [19] proposes a holistic approach for regional flood hazard risk assessment based on spatial information inducing flooding. Likewise, accurate results in flood mapping were obtained for small urban areas ([24]), large cities ([25]) and even for very large regions (entire mountainous area of China, [26]). However, for regression applications, studies focusing on flood forecasting are hardly documented. Albers et al. [27], for instance, determined the relative importance of contributing upstream discharges to the main river during flood events rather than focusing on a numerically forecasting of flash-flood events.

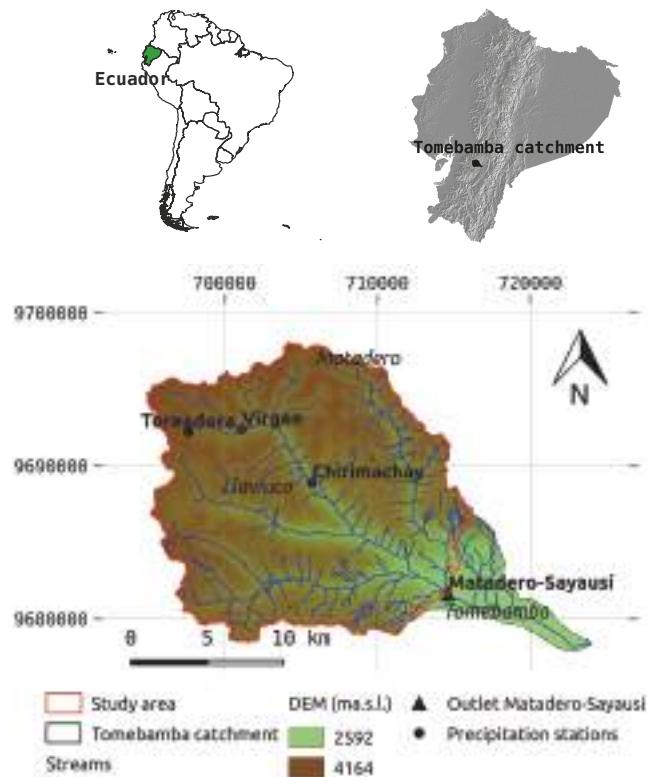
Determination of the size of the forecast horizon (lead time) is critical in time series prediction [15]. The predictive ability decreases as the number of time steps to forecast increases as result of error accumulation, accuracy reduction and lack of information [20,28]. On the other hand, identification of the regressor size (number of previous time steps from precipitation and discharge time series) aims to avoid the addition of irrelevant features that add extra noise to the training process. However, the loss of relevant features might, in some instances, also reduce prediction success [20].

In the pursuit of model parsimony, a further step is the optimization of the input composition through a feature reduction process. It can be done through a feature selection technique (sensitivity analysis) aimed to assure high accuracy, speed, robustness, efficiency and interpretability [29,30]. This technique consists of an educated retention of the most relevant features (predictors) that will reduce the model input dimension, improving its computation time without compromising its effectiveness.

The aim of this study is to develop a step-wise input data selection strategy together with a sensitivity analysis for building up an optimal forecast hydrological model based on the RF algorithm, applied to a mountain catchment. The catchment of study is representative of the Andean region of Ecuador.

## 2. Study Area and Dataset

The study area is the Tomebamba catchment towards its outlet at the Matadero-Sayausí station, which corresponds mainly to a páramo ecosystem located in the south-eastern flank of the Andean cordillera of Ecuador (Figure 1). The Tomebamba catchment is part of the Cajas National Park, which was declared by UNESCO as a World Biosphere Reserve in 2013. Elevation of the study area ranges from 2800 to 4100 m above the sea level (m.a.s.l.), and covers an approximate area of 300 km<sup>2</sup>. It drains to the upper part of the Tomebamba river (area of the catchment  $\approx$  332 km<sup>2</sup>), and lately to the Amazon river towards the Atlantic Ocean. The importance of the catchment is related to its water supplier role for domestic, agricultural and industrial purposes for the city of Cuenca (third largest city in Ecuador with more than 580,000 inhabitants), and even for hydro-power generation in the region.



**Figure 1.** Location of the Tomebamba catchment at Matadero-Sayausí outlet in the Andean cordillera of Ecuador, South America (UTM coordinates).

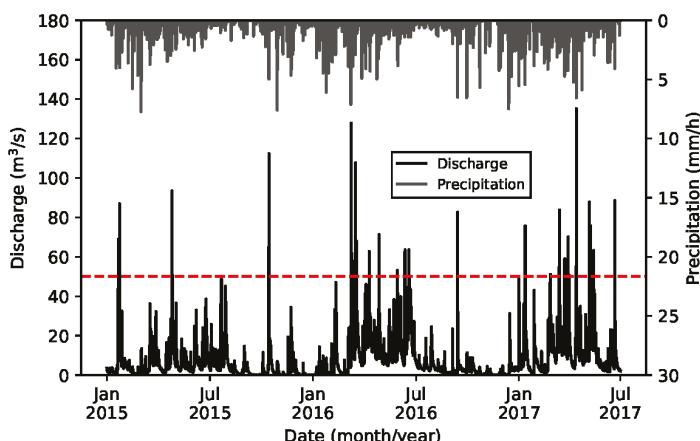
Climate at the Tomebamba catchment is governed by continental air masses from the Amazon river basin, by the seasonal shift of the Inter Tropical Convergence Zone (ITCZ), and by the cold water upwelling of the Humboldt ocean current. As a result, convective and orographic cloud formations occurs [31]. Although precipitation (in terms of volume) mainly falls as drizzle ( $\approx$ 1 mm/h), higher intensity events are also experienced in the catchment (up to 140 mm/h) [32,33].

According to the classification system of the Food and Agriculture Organization (FAO) of the United Nations [34], the prevailing soil types in the Tomebamba catchment are andosol and histosols. Furthermore, studies by [35,36] in a comparable Andean micro-catchment have revealed the dominance of interflow as the major contributor to runoff generation. This is due to the inner soil properties of water storage (andosols) and organic matter content (histosols) together with the presence of wetlands and the slope of the catchment. As a result, flash-flood events are not exclusively caused by extreme precipitation events. Non-extreme precipitation events can trigger flash-floods when the soil is saturated due to the high retention capacity of the catchment. High flows are explained by the presence of histosols, whereas in contrast, low flows generation is controlled by the slope of the catchment.

Lack of spatial information describing the main flash-flood driving forces in the region limited the variables to be used as inputs to punctual measurements (time series) of precipitation and discharge. Data comprises precipitation and runoff hourly time series for a period of 2.5 years (from January 2015 to July 2017). To account for the variability of precipitation in the catchment, precipitation time series were obtained from 3 tipping-bucket rain gauge stations (Toreadora at 3955, Virgen at 3626 and Chirimachay at 3298 m.a.s.l.), which were installed within the catchment and along its altitudinal gradient. On the other hand, discharge time series were obtained for the outlet of the study catchment, at the Matadero-Sayausí station (2693 m.a.s.l.), for which the corresponding drainage area was delineated (Figure 1).

The length of the study period was further divided for training (from January 2015 to July 2016) and validation (from July 2016 to July 2017) purposes. Mean annual precipitation volumes for the study period were 1109, 1021 and 909 mm for Chirimachay, Toreadora and Virgen stations, respectively.

Figure 2 shows the precipitation (average of 3 stations) and the discharge hourly time series for the study period. An analysis of historical discharge extreme high events in the Tomebamba catchment (from 1997 to 2017) together with local media reports from the last decade served to determine a threshold value of  $50 \text{ m}^3/\text{s}$  as an indicator of a dangerous event affecting the everyday life of the community and usually leading to a flood event in the catchment. This reference value has a return period of 4 months (calculated with WETSPRO, using peak-over-threshold values, see [37]). For the study period, 12 independent events were above this threshold value.



**Figure 2.** Precipitation (averaged for Toreadora, Virgen and Chirimachay stations) and discharge hourly time series for the study period (January 2015 to July 2017). Note the horizontal red dashed line at a discharge of  $50 \text{ m}^3/\text{s}$  (historical indicator of a flood event).

### 3. RF Technique

RF is a supervised ML algorithm that ensembles multiple decorrelated DTs [20]. A decision tree is a stochastic model that relates a response/outcome to explanatory variables or features. Each decision tree can be seen as a set of conditions, hierarchically organized, and successively applied to a dataset. Decorrelation among trees is assured by growing trees from different randomly resampled training sets (bagging technique) from the original dataset [22]. For regression applications, multiple DTs provide independent numerical predictions of the phenomenon of interest (i.e., discharge), contrary to class labels for classification. At the end, the outcome corresponds to the mean prediction of all individual trees.

Starting from the root (parent) node of a tree, and at every node, data is recursively partitioned into two self-similar child nodes by following simple rules related to the data and until a specified stop condition is reached [38]. To split a node, a random selection of features is used by the RF algorithm (instead of using all features). For this, a random component is used to resample and to select the optimal successive directions (features) for splitting the data with the purpose to obtain purer nodes than the parent. By minimizing node impurity, the collection of outcomes of a tree is the most homogeneous possible. Every terminal node comprises a simple regression model that applies in that node only. A detailed description of the method can be found in [20].

Strengths of the RF algorithm are related to its simplicity due the few parameters that need to be tuned, higher accuracy when compared to other ML techniques, and robustness as result of a bagging process [20]. It is capable of dealing with small size samples, high-dimensional spaces, and complex data structures [23]. The RF algorithm was used by means of the scikit-learn package for ML in Python® [39]. The main functions, attributes and methods employed can be found in <http://scikit-learn.org>.

#### 3.1. Algorithm

The RF algorithm for regression applications is as follows:

- (i) Construct each one of the decision tree models based on a random selection of a number of bootstrap samples ( $n\_estimators$  parameter) drawn with (or without) replacement from the training dataset. Each bootstrap is composed by a different subset (roughly two-thirds) of the dataset, in a process known as out-of-bag (OOB) [19]. The OOB technique aims to get unbiased estimates of the regression as well as to get estimates of the importance of the variables used for the tree construction process [40].
- (ii) Determine a number of features ( $max\_features$  parameter) to perform the best split decision from the total number of predictor variables of the dataset ( $n\_features$ ). The condition  $max\_features < n\_features$  ensures the nonexistence of duplicated DTs in the forest. Consequently, by assuring variety, the problem of over-fitting is avoided. Ref. [20] recommends  $max\_features = \sqrt{n\_features}$ , for regression problems.
- (iii) Split each node of each decision tree into two descendant nodes by using the best split criteria. The calculation of the best splits are chosen based on the mean squared error (MSE) for regression problems. The minimum number of samples required to split a node is controlled by the  $min\_samples\_split$  parameter.
- (iv) Grow  $n\_trees$  as much as possible (largest extent) by repeating steps 1 to 3 until a number of nodes have been reached. The optimal number of trees is reached when the  $OOB_{error}$  stops decreasing. The depth of each tree is controlled by the  $max\_depth$  and the  $min\_samples\_leaf$  parameters; where  $min\_samples\_leaf$  is the minimum number of samples required to be at a leaf node. This is aimed to reduce the structural complexity of the models in what is called pruning criteria [22].
- (v) Determine the prediction as the mean response from all regression trees [20].

### 3.2. Determination of Model Hyper-Parameters

Before using the RF algorithm, its model hyper-parameters, which basically control the structure and level of randomness of the forest [41], must be defined. For this, a random grid-search procedure was implemented to determine the optimal hyper-parameters for each model that will be built up. It consists on the assessment of the model residual mean for different combinations of the hyper-parameters on the training dataset. For this, a 3-fold cross-validation was performed to avoid overfitting. The training data was split into 3 folds; the model was iteratively fitted on the 2 folds and evaluated on the third one. Table 1 shows the ranges of the different values of the most important hyper-parameters (according to [39]) that were evaluated. From those hyper-parameters, *n\_estimators* and *max\_features* are generally agreed as the most important ones during calibration (significant impact on error rate) [19,27].

**Table 1.** Grid of the hyper-parameters used for a 3-fold cross-validation.

Hyper-Parameter	Values
<i>n_estimators</i> *	50–700
<i>max_features</i>	'auto', 'sqrt' and 'log2'
<i>min_samples_split</i>	2, 5 and 10
<i>min_samples_leaf</i>	1, 2 and 4
<i>max_depth</i> *	10–700

\* Increment of 10 units.

### 3.3. Input Data Composition

The determination of the optimal model input plays a key role in model performance since it provides the basic information about the system [29]. For a precipitation-runoff forecasting model, the input coming from precipitation at the current time alone is not sufficient for the model to perform adequately. Therefore, additional information can be derived from previous time steps of precipitation and discharge. Nevertheless, an autoregressive exogenous analysis is necessary to determine the number of lags (features) of precipitation and discharge that have a significant influence on the predicted flow.

Physically, the addition of precipitation lags aims to mimic the antecedent soil moisture conditions of the catchment, which might play a key role in flash-flood forecasting (humid regions). It accounts for the fact that during dry periods, the soil is below field capacity, and therefore, it needs additional rain water first to reach field capacity and then to generate discharge. Conversely, during wet periods, the soil is at or above field capacity and needs less water to generate discharge. As a result, simulations of runoff for dry and wet periods are characterized by underestimation and overestimation of discharge, respectively [42].

To determine the number of lags to be used, [43] proposed a qualitative analysis that relies on statistical properties such as cross-, auto- and partial-auto-correlation of the data series. This method avoids a long trial-and-error-procedure when identifying the optimal composition of the input. For discharge, the autocorrelation function (ACF) and the partial autocorrelation function (PACF) with 95% confidence levels can suggest the influencing antecedent flow patterns in the discharge at a given time [6,43,44].

For precipitation, the number of regressors can be determined either through a Pearson cross-correlation applied to the data series [43], or according to the concentration time of the catchment [44]. The above input selection procedure relies on the linear relationship between the variables; however, the effect of an additional variable is not assessed.

#### 4. Model Evaluation

Model performance or efficiency was determined by comparing the model outputs and the observed time series. For this, goodness-of-fit statistics and graphical interpretation techniques were applied together since both approaches complement each other. The shortcoming of applying only goodness-of-fit statistics is that they restrict the model assessment by representing the mean performance of the model without considering the unbalanced influence of outliers and/or extreme high (floods) or low (droughts) values [37].

##### 4.1. Goodness-of-Fit Statistics

Model residual mean ( $ME$ ) measures the average systematic difference between simulated and observed values. In contrast, the model residual variance ( $S_{EQ}^2$ ) measures the average random differences. For high number of observations, the  $MSE$  equals approximately the sum of the squared model residual mean and the model residual variance. As a result, the  $MSE$  comprises a systematic (bias in the model) and random component (after bias correction) [37].

The disadvantage of using  $MSE$ ,  $S_{EQ}^2$  and  $ME$  is their high dependence on the magnitude of variable of interest (i.e., discharge). Thus, the coefficient of efficiency also named the Nash-Sutcliffe model efficiency ( $NSE$ ), which is less sensitive to high extreme values [45], was applied to measure the overall model accuracy. The  $NSE$  is a dimensionless and scaled version of the  $MSE$ ; it is the fraction of variability in the observations explained by the model, and it can be calculated as follows:

$$NSE = \left[ 1 - \frac{\sum_{i=1}^n (Q_m(i) - Q_o(i))^2}{\sum_{i=1}^n (Q_o(i) - \bar{Q}_o)^2} \right] = \left[ 1 - \frac{MSE}{S_{Q_o}^2} \right] \quad (1)$$

where  $Q_o$  is the mean observations value. The  $NSE$  ranges between  $-\infty$  and 1.0, being  $NSE = 1$  the optimal value.

The systematic error, which is the objective function, can be minimized through calibration. On the contrary, the random component cannot be removed since it is related to the inherent stochastic nature of the inputs.

Although ideal, the use of target functions designed for extreme high flows such as the mean peak difference (see [46]) was not possible due to the relatively small dataset available for the study catchment. The reason is the poor training of the trees in the forest when the objective function is specialized in extreme high values. Therefore, to use target functions larger datasets are required to capture a more complete spectrum of extreme events. Nonetheless, a solution to this issue relied in training the model for all flows, enriching the input of the model with additional information specifically aimed to improve the prediction of extreme high flows, and then assessing the efficiency of the model via means of graphical analyses focused on extreme high values (floods).

##### Graphical Analysis

The need to perform a more complete evaluation of model performance (analysis of extreme high values) is related to the fact that model residuals ( $ME$  and  $S_{EQ}^2$ ) increase with higher flows. For this, a Box-Cox transformation of discharges was applied; it was calculated with the following equation:

$$Box - Cox(q) = \frac{q^\lambda - 1}{\lambda} \quad (2)$$

where  $q$  is discharge and the parameter  $\lambda$  can be calibrated graphically until reaching homoscedasticity in the residuals (constant standard deviation). However, according to [37], a value of lambda = 0.25 is commonly used for runoff, and thus it was adopted for this study.

Another issue related to model residuals is their common serial dependence on scales (time step) and flow magnitudes—it will be higher for shorter time steps [37]. For peak flows (floods), the serial dependence is strong for short time steps (i.e., hourly scales or smaller than the recession constant of the quickest sub-flow component). The evaluation of flows occurring at all time steps will imply a higher representativeness for low flows; therefore, the selection of nearly independent observations is necessary. Independent events were obtained by splitting the discharge time series in events and using one value per event (technique developed by [37]).

Finally, the assessment of a model under extreme conditions relied on an indirect indicator, the flow frequency distribution for extreme high values. The analysis focused on the behavior of the distribution towards the tail (i.e., higher values for peak frequency distribution). A comparison of discharge values with a visual inspection of the shape of the tail for both historical observations and simulations was applied to conclude whether model performance is acceptable for extreme conditions.

#### 4.2. Feature Selection Analysis

Model parsimony was approached via a reduction of the input dimension using a procedure known as feature selection. It is aimed to shorten computation times by determining the relative importance of each of the features of the input and trimming off the less important ones. In some cases, feature selection even improves model accuracy [47]. Different techniques for feature selection are available; e.g., based on a variance sensitivity analysis, based on univariate statistical tests, recursive elimination, among others.

For this study, the variance sensitivity analysis was applied by following a simple procedure introduced by [48]. It measures the variance of the output produced by a single feature without considering interactions between features. The selection criteria rely on the fact that a relevant feature to the model will produce a higher output variance. The variance ( $V_k$ ) produced by a single feature, and therefore, its relative importance ( $R_k$ ) can be calculated as follows:

$$V_k = \frac{\sum_{j=1}^L (\hat{y}_{t-k}(j) - \overline{\hat{y}_{t-k}(j)})^2}{L - 1} \quad (3)$$

$$R_k = \frac{V_k}{\sum_{i=1}^m V_i} \times 100 \quad (4)$$

where,  $\hat{y}_{t-k}$  is the model output obtained by holding all  $m$  input features at their average values except  $\hat{y}_{t-k}$ , which varies according to the sample or time step, along the interval  $j \in \{1, \dots, L\}$ .

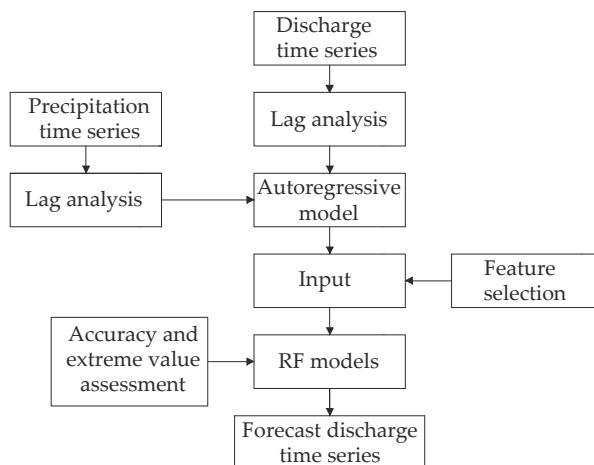
Finally, based on the  $R_k$ -values obtained, the selection principle was to retain a number of features accounting for at least 80 % of the total relative importance. The remaining features can be considered unimportant and therefore removed.

#### 5. Step-Wise Methodology

The step-wise methodology proposed in this study (Figure 3) consists in setting up several RF models to forecast discharge for a specified forecast horizon until finding the optimal or parsimonious one in terms of model efficiency. Special attention is given to the forecast of extreme high values (floods) via an analysis of their flow frequency distribution. At every step, a new model is composed by a particular input data (time series) based on the information of precipitation and discharge, and according to a lag analysis aimed to include only relevant information.

For this purpose, an autoregressive model, whose input is merely composed by discharge lags, was defined as the base model. The dimension of the input of the base model is then gradually increased by adding extra information such as precipitation observations at the current time and lags (to mimic soil moisture conditions in the catchment). Model hyper-parameters are tuned for every new model (input data composition scenario).

The outcome of the RF model is the prediction of discharge at a certain time, which lately forms a discharge time series for the specified forecast horizon. The last step, whose objective is to derive parsimonious models, is to reduce the dimension of the input based on a sensitivity analysis that determines the most relevant precipitation features accounting for the 80% of the output's variance. The step-wise approach proposed in this study is firstly used for building up a 4-h discharge forecasting model, and later on, the arising results will serve to develop models with longer forecast horizons of varying time (8, 12, 18 and 24 h).



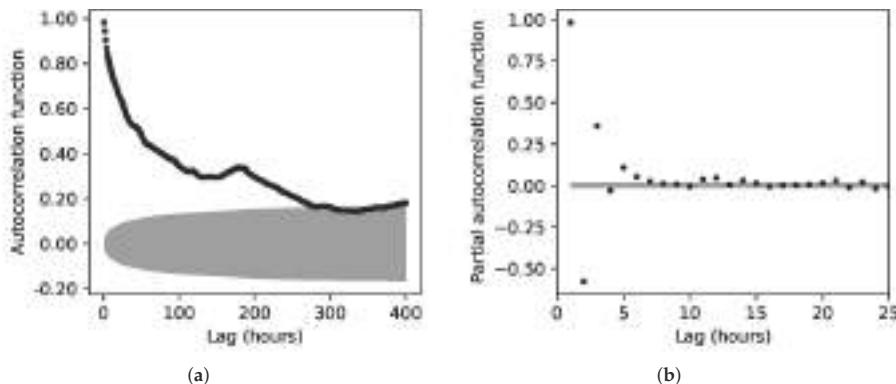
**Figure 3.** Scheme of the step-wise methodology for developing precipitation-runoff forecasting models.

## 6. Results and Discussion

The following subsections, unless specifically mentioned, correspond to a 4-h discharge forecasting model.

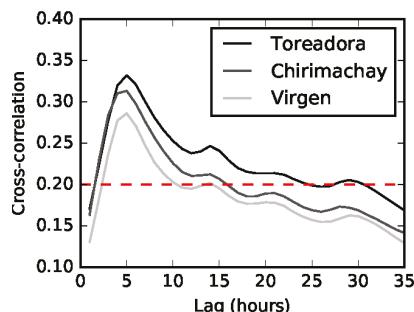
### 6.1. Determination of the Number of Discharge and Precipitation Lags

Figure 4 shows the ACF and PACF plots, respectively, from which the number of discharge lags were determined. The ACF and the corresponding 95% confidence band was estimated from lag 1 to lag 400 (h), and the highest autocorrelation occurred at the first lag. A significant correlation was revealed up to lag 300 (around 13 days). Thereafter, the correlation fell within the confidence band (Figure 4a). The systematic ACF decay demonstrated the presence of a dominant autoregressive process. Similarly, the PACF and its 95% confidence levels were estimated from lag 1 to 25. The PACF revealed a significant correlation up to lag 8. The dominance of the autoregressive process over the moving-average is proved by the rapid decay of the PACF (Figure 4b). As a result, it seemed reasonable to include discharges up to 8 lags (h) as additional inputs.



**Figure 4.** (a) Autocorrelation function (ACF) and (b) Partial autocorrelation function (PACF) of the Matadero-Sayausí discharge series. Gray hatch indicates the 95% confidence band.

Figure 5 illustrates the Pearson's cross-correlation between each precipitation station and the discharge time series to determine the number of precipitation lags to be used. Results revealed the highest correlation, for all stations, at lag 4 (maximum correlation of 0.3323 for Toreadora). In contrast, the concentration time of the catchment was estimated in 2.3, 3.4, 5.2 and 5.9 h according to the equations of Kirpich, Giandotti, Ven Te Chow and Temez, respectively (a summary of the equations can be found in [49]). The average concentration time (4.3 h), according to the mentioned equations, matches with the number of lags determined by the cross-correlation analysis.



**Figure 5.** Pearson cross-correlation comparison between the Toreadora (3955 m.a.s.l.), Virgen (3626 m.a.s.l.) and Chirimachay (3298 m.a.s.l.) precipitation stations and the Matadero-Sayausí (2693 m.a.s.l.) discharge station. Note the horizontal line at a cross-correlation of 0.20.

## 6.2. Base Model: Discharge Lags as the Sole Input

The use of 8 discharge lags as the input of the base model (model A) was confirmed by a sequential process that gradually added one lag at the time until the NSE-values calculated for the training period stopped significantly (0.005) improving. Although the addition of more than 8 lags resulted in a slight increase of the NSE-values in the training period, the NSE-values in the validation period started deteriorating (as a result of overfitting).

Model efficiencies of the base model, whose input consisted of 8 features, obtained NSE-values of 0.880 and 0.652 for the training and validation periods, respectively. Moreover, according to the sensitivity analysis, the relative importance of each of the discharge lags decreased slightly and proportionally from 13.6 % (lag<sub>-1</sub>) to 11.6% (lag<sub>-8</sub>).

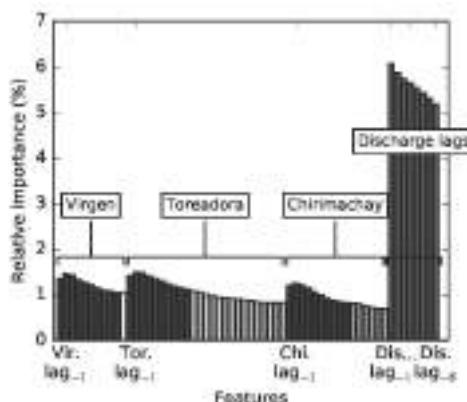
### 6.3. Precipitation Lags as Additional Inputs

To improve the base model, its input was enriched with precipitation information coming from three stations (Toreadora, Virgen and Chirimachay) from which data were available. The number of lags was determined according to the cross-correlation between each precipitation station and the discharge time series. For this, similar to [29,43], a cross-correlation threshold of 0.2 was selected, indicating that 24, 10 and 15 lags should be used for Toreadora, Virgen and Chirimachay stations, respectively.

The input of this new model (model B) was composed by precipitation information at the current time from 3 stations, their corresponding lags and discharge lags (60 features in total). Results indicated NSE-values of 0.954 and 0.758 for the training and validation periods, respectively.

### 6.4. Feature Selection

Figure 6 shows the relative importance of each of the features of model B. Results clearly show the predominance of the discharge information over precipitation. The total relative importance of the precipitation information was 44.8%. Based on the sensitivity analysis, the contribution of the three rainfall stations was 26.9, 13.5 and 14.8% for Toreadora, Virgen and Chirimachay, respectively. No relation was found between the relative importance of the precipitation station and its altitude nor distance to the outlet.

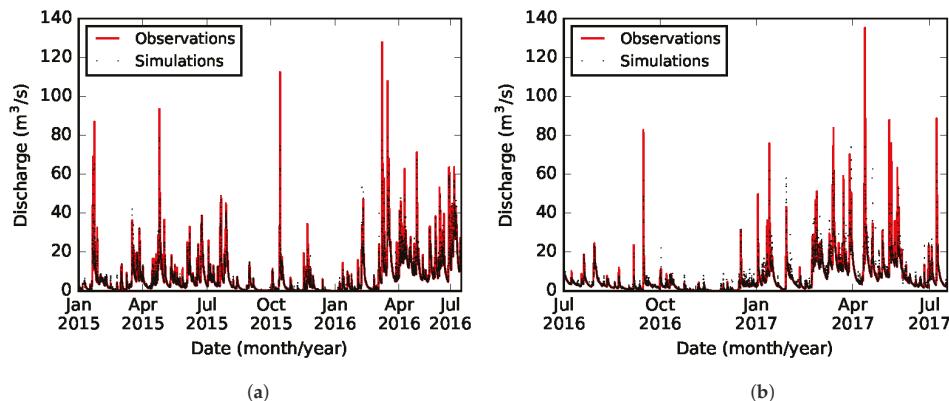


**Figure 6.** Feature relative importance of the 4-h discharge forecasting model B. Darkest bars indicate the features selected for a reduced version of the input (model C).

As a final stage, precipitation at the current time and 9 lags for all stations (from lag<sub>-1</sub> to lag<sub>-9</sub>) were selected since they accounted for the 80.36 % of the total relative importance of model B. As a result, 38 features in total were used as inputs of model C (see darkest bars in Figure 6). Efficiency of model C slightly improved (0.018) when compared to model B. Results indicated NSE-values of 0.972 and 0.761 for the training and validation periods, respectively. Figure 7 presents the model results for both the training and validation periods. Moreover, a comparison between the results of models B and C showed a correlation coefficient ( $R^2$ ) of 0.996. Both the slight change of model efficiencies and the comparison of results suggested that 36.7 % of the features were successfully trimmed off, achieving a parsimonious model.

In contrast, to study the usefulness of the addition of precipitation information, the lags of each station were gradually and simultaneously included in the input until reaching the best NSE-values for the training period. Results proved that only 4 lags per station were enough to achieve the best model performance (0.973). Results obtained NSE-values of 0.973 and 0.764 for the training and validation periods, respectively. Only an improvement of 0.001 of efficiency (training period) was obtained when

compared to model C. Thus, the criteria of including precipitation information until reaching the 80% of the relative importance was confirmed.



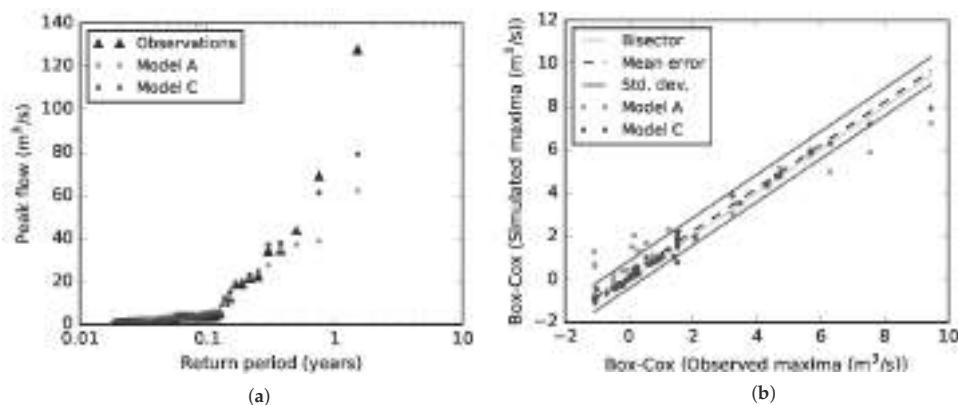
**Figure 7.** Model results of the parsimonious 4-h discharge forecasting (model C) (a) Training period, from January-2015 to July-2016. (b). Validation period, from July-2016 to July-2017.

### 6.5. Graphical Analysis

To complement goodness-of-fit statistics, Figure 8a shows the empirical extreme value distribution of peak flows for both observations and simulations (models A and C). The use of a pure autoregressive process (model A) underestimated systematically, to a greater degree, peak flows for 0.3-year return period onwards when compared to model C.

Additionally, Figure 8b shows the correlation between the predictions of models A and C (vertical axis) and the observed flows (horizontal axis) for high flows. The mean error and the standard deviation correspond to the results of model C. Model residuals are represented by the horizontal and vertical differences between each point and the bisector line. The dependence of the standard deviation on the flow magnitude was disrupted (constant standard deviation) after applying the Box-Cox transformation with a  $\lambda$ -value of 0.25. Predictions of model A indicate higher scatter (higher standard deviation of peak flow deviations from the bisector line) and higher bias (systematically lower mean peak flows) when compared to model C. Consequently, peak flows were systematically more underestimated by model A than model C. The improvement in the prediction of peak flows when precipitation data were included as additional inputs is therefore evident—it is considerably more significant for extreme high values. For instance, the 4-h forecast of a particular peak discharge of  $69 \text{ m}^3/\text{s}$  was improved from 39 (model A) to 61  $\text{m}^3/\text{s}$  (model C).

Overall assessment of the 4-h forecasting model indicates a good match between observations and simulations for flows up to  $60 \text{ m}^3/\text{s}$  (being  $7.1 \text{ m}^3/\text{s}$  the mean discharge of the time series). This confirms the validity of the derived model for the forecasting of discharges historically indicating floods in the catchment (threshold value of  $50 \text{ m}^3/\text{s}$ ). The fact that flows above this threshold were correctly simulated determined the sufficiency of 2.5 years for calibration/validation when the objective is to provide an alarm indicating a flood risk in the catchment.



**Figure 8.** (a) Empirical extreme value distribution of peak flows; (b) Comparison of nearly independent peak flow maxima.

#### 6.6. Forecasting Models of 8, 12, 18 and 24 h

The step-wise methodology applied for building up a 4-h forecasting model was further tested for longer prediction horizons of 8, 12, 18 and 24 h. Table 2 resumes their corresponding input compositions, which show that the input dimension of the best model increased (precipitation lags) accordingly to the increase of the forecast horizon analyzed. More than 8 discharge lags were only needed for the 24-h forecasting model.

**Table 2.** Input data composition and model efficiencies for forecasting models and their parsimonious versions for prediction horizons of 4, 8, 12, 18 and 24 h.

Forecast Horizon [h]	Discharge Lags	Toreadora Lags	Virgen Lags	Chirimachay Lags	Total Features	NSE Training	NSE Validation
4	8	24	10	15	60	0.954	0.758
4 *	8	9	9	9	38	0.972	0.761
8	8	32	19	23	85	0.868	0.581
8 *	8	15	15	15	56	0.867	0.580
12	8	36	23	27	98	0.828	0.506
12 *	8	18	18	18	65	0.829	0.503
18	8	42	29	33	115	0.772	0.442
18 *	8	21	21	21	74	0.771	0.439
24	15	48	35	39	140	0.772	0.385
24 *	15	21	21	21	81	0.767	0.384

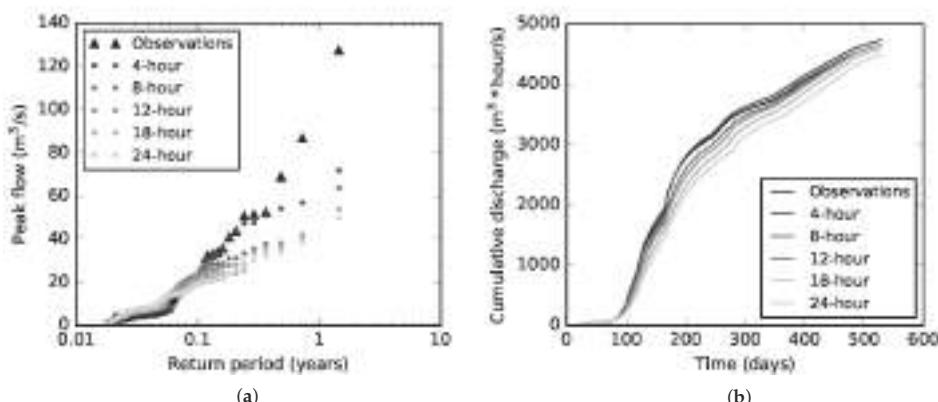
\* Parsimonious version.

Feature selection based on the variance analysis was successfully performed for all forecasting models when a fixed cross-correlation threshold of 0.2 and a target of cumulative 80 % of relative importance were used for all forecast horizons. The percentage of reduction of features without compromising model performance were 36.7, 34.1, 33.7, 35.7 and 42.1% for forecasting models of 4, 8, 12, 18 and 24 h, respectively. Additionally, a comparison between model results of the full and optimal models showed correlation coefficients ( $R^2$ ) of 0.985, 0.996, 0.996, and 0.997 for forecast horizons of 8, 12, 18 and 24 h.

Model efficiencies in terms of NSE-values for the training and validation periods are shown in Table 2. Results proved that the ability of a model to forecast discharge decreases as the forecast horizon increases. Regarding the effectiveness of the feature selection analysis, only a slight change

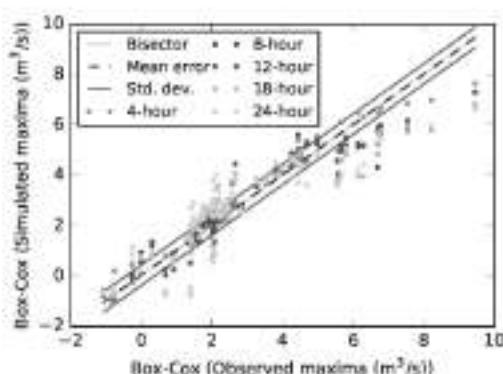
(maximum 0.003) was observed in the NSE-values for the validation period; in some cases feature selection improved model efficiencies.

Figure 9a compares the empirical peak flow extreme value distribution of all forecast horizons (4, 8, 12, 18 and 24 h) for both observations and simulations. Only the optimal models were used for this analysis. The underestimation of extreme values towards the upper tail of the distribution became stronger as the forecast horizon increased. The lower bias of shorter forecast horizons in the empirical distribution is also reflected in the cumulative hydrograph volume (Figure 9b).



**Figure 9.** (a) Empirical extreme value distribution of peak flows, and (b) Comparison of cumulative flow volumes for forecasting models of 4, 8, 12, 18 and 24 h.

Similarly, Figure 10 shows higher scatter and bias for longer forecast horizons, which resulted in higher forecast errors when the forecast horizon increased. Strong discharge underestimations (up to 300% for the 24-h forecasting model) are explained by the simplicity of precipitation-runoff forecasting models, which lack of relevant information describing the flash-flood generation process in mountainous areas. Particularly, in páramo ecosystems, disregard of directly measured soil moisture information limits the forecasting of extreme high and low flows. As a result, the problem of underestimation of low flows and overestimation of high flows was observed in the peak flow empirical extreme value distribution.



**Figure 10.** Comparison of nearly independent peak flow maxima for forecasting models of 4, 8, 12, 18 and 24 h.

Another factor that compromises the forecasting of extreme high flows is the insufficient representation of the spatial variability of precipitation in the catchment with only three rain gauges stations. Precipitation events might occur in areas out of the coverage of rain gauges, specifically, rain formation processes that are driven by local topography.

Besides the lack of relevant information feeding the models, the use of a relatively small dataset (2.5 years) for training and validation might also reduce the ability of the RF models to forecast discharge. This is due to the incomplete spectrum of discharge magnitudes captured by a small dataset, specially when extreme high flows are of interest. As a result, the RF models will not perform well for data points beyond the scope of the training dataset due to the average of the results of each tree in the forest and as well as the rules established for partitioning data during the training process. Results of the combined effect of the lack of relevant information and the dataset limitation can be also seen in Figure 9a, where higher errors were found for longer forecast horizons.

## 7. Conclusions

The development of an adequate runoff and flash-flood forecasting model is needed due to the susceptibility of lowland areas to the catastrophic socio-economic impacts of flash-floods. However, the development of such model is a complex procedure that deals with non-linear stream flow generation processes. Specifically, the use of precipitation-runoff forecasting models is targeted, as an efficient solution, for regions with lack of key-spatial information explaining the flash-flood generation process. When a model is approached through ML techniques, small datasets, high dimensionality and the difficulty to identify the importance of input features limit the proper construction of a model, and consequently, model performance optimization. Hence, to account for this issues, the use of RF models was proposed in this study to assess short-term discharge forecasting models of varying duration time (4, 8, 12, 18 and 24 h) for a mountainous region with data limitation issues.

The methodology proposed in this study have improved the current knowledge of the precipitation-runoff relations in terms of prediction over a catchment located at a high altitude. The RF ability to predict extreme values decreased as the forecast horizon increased according to goodness-of-fit statistics together with graphical analysis. Additionally, an extreme value analysis served to prove the improvement in the prediction of extreme peak values as a result of the addition of precipitation information to pure autoregressive models.

Input dimensionality reduction, applied through a feature selection method based on a sensitivity analysis, was performed for all forecasting scenarios by retaining the features accounting for 80% of the model's variance. Significant correlation coefficients between the model results of the full and its parsimonious version, and a slight difference in NSE-values proved that the selection of the most important features was successfully achieved. Feature selection aimed to reduce the complexity of the model and to identify the processes involved in discharge prediction. At the same time, computation times were optimized.

The approach hereby proposed have the potential to be applied in different catchments sizes; however, a maximum extension of around 1000 km<sup>2</sup> is suggested. This value intends to cover all the catchments associated with an Andean community. For larger catchments, the use of spatial information is encouraged since other relevant factors to the flash-flood generation process (e.g., topography, soil types and land uses) can be correctly represented by the most common remote sensing products. This is not the case for the Andean region, where only exceptionally fine detail, which is not feasible to obtain, would improve the model performance of RF models.

Emerging advances in ML techniques, specifically the use of the RF algorithm, have shown to serve as a powerful tool in short-term runoff forecasting. However, the use of the RF technique has been hardly documented on flood hazard assessment and particularly flash-flood forecasting. Further exploration of this technique for flash flood forecasting is therefore encouraged. Expansion of the rain gauge network or the inclusion of remote sensing imagery together with ground validation

is suggested to improve the representativeness of precipitation in mountain catchments. Likewise, addition of soil moisture measurements is proposed for enriching the model to further evaluation.

**Author Contributions:** Methodology, P.M., J.O.-A. and R.C. Modeling and Formal Analyses, P.M. Writing—Original Draft Preparation, P.M. Writing—Review & Editing, J.O.-A., P.W. and R.C. Supervision, J.O.-A., R.C. and P.W. Conceptualization, R.C.

**Funding:** P.M. was funded by a Flemish Interuniversity Council (VLIR-OUS) scholarship. This research was part of the project “Identificación de los procesos hidrometeorológicos que desencadenan crecidas extremas en la ciudad de Cuenca” funded by the Dirección de Investigación of Universidad de Cuenca (DIUC) and the Empresa Pública Municipal de Telecomunicaciones, Agua Potable, Alcantarillado y Saneamiento de Cuenca (ETAPA-EP).

**Acknowledgments:** We are grateful to the staff and students that contributed to the hydrometeorological monitoring. We would like to acknowledge the information provided by ETAPA-EP for the development of this study. Special thanks are given to the Integrated Research on Disaster Risk (IRDR) Young Scientists Programme.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine learning
RF	Random forest
ANN	Artificial neural network
SVM	Super vector machine
ME	Mean error
MSE	Mean square error
NSE	Nash-Sutcliffe efficiency

## References

1. Stefanidis, S.; Stathis, D. Assessment of flood hazard based on natural and anthropogenic factors using analytic hierarchy process (AHP). *Nat. Hazards* **2013**, *68*, 569–585. [[CrossRef](#)]
2. Min, S.K.; Zhang, X.; Zwiers, F.W.; Hegerl, G.C. Human contribution to more-intense precipitation extremes. *Nature* **2011**, *470*, 378–381. [[CrossRef](#)] [[PubMed](#)]
3. Fi-John, C.; Hwang, Y.Y. A self-organization algorithm for real-time flood forecast. *Hydrol. Process.* **1999**, *13*, 123–138.
4. D’Ercole, R.; Trujillo, M. *Amenazas, Vulnerabilidad, Capacidades y Riesgo en el Ecuador*; Coopi-IRD-Oxfam: Quito, Ecuador, 2002.
5. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models part I—A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [[CrossRef](#)]
6. Wang, W.; Gelder, P.H.A.J.M.V.; Vrijling, J.K.; Ma, J. Forecasting daily streamflow using hybrid ANN models. *J. Hydrol.* **2006**, *324*, 383–399. [[CrossRef](#)]
7. Galelli, S.; Castelletti, A. Assessing the predictive capability of randomized tree-based ensembles in streamflow modelling. *Hydrol. Earth Syst. Sci.* **2013**, *17*, 2669–2684. [[CrossRef](#)]
8. Braud, I.; Ayral, P.A.; Bouvier, C.; Branger, F.; Delrieu, G.; Dramais, G.; Le Coz, J.; Leblois, E.; Nord, G.; Vandervaere, J.P. Advances in flash floods understanding and modelling derived from the FloodScale project in South-East France. In Proceedings of the 3rd European Conference on Flood Risk Management, Innovation, Implementation, Integration (FLOODrisk 2016), Lyon, France, 18–20 October 2016; Volume 7, p. 04005.
9. Ruin, I.; Creutin, J.D.; Anquetin, S.; Lutoff, C. Human exposure to flash floods—Relation between flood parameters and human vulnerability during a storm of September 2002 in Southern France. *J. Hydrol.* **2008**, *361*, 199–213. [[CrossRef](#)]
10. Buytaert, W.; Celleri, R.; Willems, P.; De Bievre, B.; Wyseure, G. Spatial and temporal rainfall variability in mountainous areas: A case study from the south Ecuadorian Andes. *J. Hydrol.* **2006**, *329*, 413–421. [[CrossRef](#)]

11. Celleri, R.; Willems, P.; Buytaert, W.; Feyen, J. Space-time rainfall variability in the Paute Basin, Ecuadorian Andes. *Hydrol. Process.* **2007**, *21*, 3316–3327. [[CrossRef](#)]
12. Espinoza Villar, J.C.; Ronchail, J.; Guyot, J.L.; Cochonneau, G.; Naziano, F.; Lavado, W.; De Oliveira, E.; Pombosa, R.; Vauchel, P. Spatio-temporal rainfall variability in the Amazon basin countries (Brazil, Peru, Bolivia, Colombia, and Ecuador). *Int. J. Climatol.* **2009**, *29*, 1574–1594. [[CrossRef](#)]
13. Dinerstein, E.; Graham, D.J.; Olsen, D.M. *Una Evaluación del Estado de Conservación de las Eco-Regiones Terrestres de América Latina y el Caribe*; Banco Mundial: Washington, DC, USA, 1995.
14. Rossenaar, A.; Hofstede, R.G.M. Effects of burning and grazing on root biomass in the páramo ecosystem. In *Páramo: An Andean Ecosystem under Human Influence*; Academic Press: London, UK, 1992; pp. 211–213.
15. Bontempi, G.; Taieb, S.B.; Le Borgne, Y.A. *Machine Learning Strategies for Time Series Forecasting*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 62–77.
16. Jin, L.; Kuang, X.; Huang, H.; Qin, Z.; Wang, Y. Study on the Overfitting of the Artificial Neural Network Forecasting Model. *Acta Meteorol. Sin.* **2005**, *19*, 216–225.
17. Martens, D.; De Backer, M.; Haesen, R.; Vanthienen, J.; Snoeck, M.; Baesens, B. Classification with ant colony optimization. *IEEE Trans. Evol. Comput.* **2007**, *11*, 651–665. [[CrossRef](#)]
18. Kubal, C.; Haase, D.; Meyer, V.; Scheuer, S. Integrated urban flood risk assessment-adapting a multicriteria approach to a city. *Nat. Hazards Earth Syst. Sci.* **2009**, *9*, 1881–1895. [[CrossRef](#)]
19. Wang, Z.; Lai, C.; Chen, X.; Yang, B.; Zhao, S.; Bai, X. Flood hazard risk assessment model based on random forest. *J. Hydrol.* **2015**, *527*, 1130–1141. [[CrossRef](#)]
20. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
21. Kühnlein, M.; Appelhans, T.; Thies, B.; Nauss, T. Improving the accuracy of rainfall rates from optical satellite sensors with machine learning—A random forests-based approach applied to MSG SEVIRI. *Remote Sens. Environ.* **2014**, *141*, 129–143. [[CrossRef](#)]
22. Rodriguez-Galiano, V.; Mendes, M.P.; Garcia-Soldado, M.J.; Chica-Olmo, M.; Ribeiro, L. Predictive modeling of groundwater nitrate pollution using Random Forest and multisource variables related to intrinsic and specific vulnerability: A case study in an agricultural setting (Southern Spain). *Sci. Total Environ.* **2014**, *476*, 189–206. [[CrossRef](#)] [[PubMed](#)]
23. Biau, G.; Scornet, E. A random forest guided tour. *Test* **2016**, *25*, 197–227. [[CrossRef](#)]
24. Feng, Q.; Liu, J.; Gong, J. Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier—A case of Yuyao, China. *Water* **2015**, *7*, 1437–1455. [[CrossRef](#)]
25. Lee, S.; Kim, J.C.; Jung, H.S.; Lee, M.J.; Lee, S. Spatial prediction of flood susceptibility using random-forest and boosted-tree models in Seoul metropolitan city, Korea. *Geomat. Nat. Hazards Risk* **2017**, *8*, 1185–1203. [[CrossRef](#)]
26. Zhao, G.; Pang, B.; Xu, Z.; Yue, J.; Tu, T. Mapping flood susceptibility in mountainous areas on a national scale in China. *Sci. Total Environ.* **2018**, *615*, 1133–1142. [[CrossRef](#)] [[PubMed](#)]
27. Albers, S.J.; Déry, S.J.; Petticrew, E.L. Flooding in the Nechozo River Basin of Canada: A random forest modeling approach to flood analysis in a regulated reservoir system. *Can. Water Resour. J.* **2016**, *41*, 250–260. [[CrossRef](#)]
28. Sorjamaa, A.; Hao, J.; Reyhani, N.; Ji, Y.; Lendasse, A. Methodology for long-term prediction of time series. *Neurocomputing* **2007**, *70*, 2861–2869. [[CrossRef](#)]
29. Wang, W.C.; Chau, K.W.; Cheng, C.T.; Qiu, L. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *J. Hydrol.* **2009**, *374*, 294–306. [[CrossRef](#)]
30. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
31. Bendix, J.; Rollenbeck, R.; Göttlicher, D.; Cermak, J. Cloud occurrence and cloud properties in Ecuador. *Clim. Res.* **2006**, *30*, 133–147. [[CrossRef](#)]
32. Muñoz, P.; Céller, R.; Feyen, J. Effect of the Resolution of Tipping-Bucket Rain Gauge and Calculation Method on Rainfall Intensities in an Andean Mountain Gradient. *Water* **2016**, *8*, 534. [[CrossRef](#)]
33. Orellana-Alvear, J.; Céller, R.; Rollenbeck, R.; Bendix, J. Analysis of Rain Types and Their Z–R Relationships at Different Locations in the High Andes of Southern Ecuador. *J. Appl. Meteorol. Climatol.* **2017**, *56*, 3065–3080. [[CrossRef](#)]
34. IUSS Working Group WRB. *World Reference Base for Soil Resources 2014*; FAO: Rome, Italy, 2014.
35. Mosquera, G.M.; Lazo, P.X.; Céller, R.; Wilcox, B.P.; Crespo, P. Runoff from tropical alpine grasslands increases with areal extent of wetlands. *Catena* **2015**, *125*, 120–128. [[CrossRef](#)]

36. Mosquera, G.M.; Céllerí, R.; Lazo, P.X.; Vaché, K.B.; Perakis, S.S.; Crespo, P. Combined Use of Isotopic and Hydrometric Data to Conceptualize Ecohydrological Processes in a High-Elevation Tropical Ecosystem. *Hydrol. Process.* **2016**, *30*, 2930–2947. [[CrossRef](#)]
37. Willems, P. A time series tool to support the multi-criteria performance evaluation of rainfall-runoff models. *Environ. Model. Softw.* **2009**, *24*, 311–321. [[CrossRef](#)]
38. Breiman, L. *Classification and Regression Trees*; Routledge: Routledge, UK, 2017.
39. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
40. Boulesteix, A.L.; Janitza, S.; Kruppa, J.; König, I.R. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2*, 493–507. [[CrossRef](#)]
41. Probst, P.; Wright, M.; Boulesteix, A.L. Hyperparameters and Tuning Strategies for Random Forest. *arXiv* **2018**, arxiv:1804.03515.
42. Willems, P. Parsimonious rainfall-runoff model construction supported by time series processing and validation of hydrological extremes—Part 1: Step-wise model-structure identification and calibration approach. *J. Hydrol.* **2014**, *510*, 578–590. [[CrossRef](#)]
43. Sudheer, K.P.; Gosain, A.K.; Ramasastri, K.S. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrol. Process.* **2002**, *16*, 1325–1330. [[CrossRef](#)]
44. Wu, C.L.; Chau, K.W. Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis. *J. Hydrol.* **2011**, *399*, 394–409. [[CrossRef](#)]
45. Moriasi, D.N.; Arnold, J.G.; Van Liew, M.W.; Bingner, R.L.; Harmel, R.D.; Veith, T.L. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Trans. ASME* **2007**, *50*, 885–900.
46. Peleg, N.; Gvirtzman, H. Groundwater flow modeling of two-levels perched karstic leaking aquifers as a tool for estimating recharge and hydraulic parameters. *J. Hydrol.* **2010**, *388*, 13–27. [[CrossRef](#)]
47. Tang, Y.; Reed, P.; Werkhoven, K.V.; Wagener, T. Advancing the identification and evaluation of distributed rainfall&hyphen;runoff models using global sensitivity analysis. *Water Resour. Res.* **2007**, *43*, 1–14. [[CrossRef](#)]
48. Cortez, P. Sensitivity Analysis for Time Lag Selection to Forecast Seasonal Time Series using Neural Networks and Support Vector Machines. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8. [[CrossRef](#)]
49. de Almeida, I.K.; Almeida, A.K.; Anache, J.A.A.; Steffen, J.L.; Sobrinho, T.A. Estimation on time of concentration of overland flow in watersheds: A review. *Geociências* **2014**, *33*, 661–671.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation

Caihong Hu <sup>1</sup>, Qiang Wu <sup>1</sup>, Hui Li <sup>2</sup>, Shengqi Jian <sup>1</sup>, Nan Li <sup>1</sup> and Zhengzheng Lou <sup>2,\*</sup>

- <sup>1</sup> School of Water Conservancy and Environment, Zhengzhou University, Zhengzhou 450001, China; hucaihong@zzu.edu.cn (C.H.); wuqianghlj@163.com (Q.W.); jiansq@zzu.edu.cn (S.J.); linan322@sina.com (N.L.)  
<sup>2</sup> School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China; iehli@gs.zzu.edu.cn  
\* Correspondence: iezzlou@zzu.edu.cn

Received: 31 August 2018; Accepted: 25 October 2018; Published: 30 October 2018

**Abstract:** Considering the high random and non-static property of the rainfall-runoff process, lots of models are being developed in order to learn about such a complex phenomenon. Recently, Machine learning techniques such as the Artificial Neural Network (ANN) and other networks have been extensively used by hydrologists for rainfall-runoff modelling as well as for other fields of hydrology. However, deep learning methods such as the state-of-the-art for LSTM networks are little studied in hydrological sequence time-series predictions. We deployed ANN and LSTM network models for simulating the rainfall-runoff process based on flood events from 1971 to 2013 in Fen River basin monitored through 14 rainfall stations and one hydrologic station in the catchment. The experimental data were from 98 rainfall-runoff events in this period. In between 86 rainfall-runoff events were used as training set, and the rest were used as test set. The results show that the two networks are all suitable for rainfall-runoff models and better than conceptual and physical based models. LSTM models outperform the ANN models with the values of  $R^2$  and NSE beyond 0.9, respectively. Considering different lead time modelling the LSTM model is also more stable than ANN model holding better simulation performance. The special units of forget gate makes LSTM model better simulation and more intelligent than ANN model. In this study, we want to propose new data-driven methods for flood forecasting.

**Keywords:** LSTM; rainfall-runoff; flood events

---

## 1. Introduction

Flooding always carries a lot of debris and waste like dead animal bodies and hazardous materials. The debris could make serious threats to mankind's health and could destroy reservoirs and roads worsening the situation. The best way to cope with these issues is to build flood management systems for the decision-making process of critical situations [1,2]. In hydrological processes, rainfall is taken major components and decided the drought or flooding events. Recently, there are mainly three types of models for simulating the relationship of rainfall and runoff [3,4]:conceptual models, physical-based models and black box models. A conceptual model is a representation of a system, made of the composition of concepts which are used to help us to know, understand, or simulate a subject the model represents [5]. A physical-based model is a smaller or larger physical copy of an object to study hydrological process [6]. A black box model is a system which can be viewed in terms of its inputs and outputs without any knowledge of its internal working [7].

With accurate modelling of rainfall-runoff dynamics, it could not only provide a flood warning to reduce hazards but also enhance proper reservoirs management during the drought periods. However, it is difficult to fully understand the relationship between precipitation and runoff. It is because of temporal and spatial variability of basin characteristics, rainfall, coverage of vegetation, as well as

factors in the rainfall-runoff process such as physical-based distributed hydrological model. Therefore, rainfall-runoff modelling is a hot field of study in hydrology research [8].

Among these three types of models, the conceptual and physical maybe the best two models to understand the process of rainfall-runoff. While these models also need more basin parameters like soil moisture, soil type, slope, shape, topography, temperature, evapotranspiration. The different watershed parameters also contain very complex relationships to construct these models [9]. Besides, In the rural region it is hard to get these watershed parameters. Therefore, black models have been increasingly emphasized during these years again [10].

These black box models are used more and more as the data-driven techniques are developing [11]. The Artificial Neural Networks (ANN), one of the data-driven techniques, have been widely used in hydrology as an alternative to physical-based and conceptual models [12,13]. These ANN techniques are based on artificial intelligence (AI), which is among the most famous skills in recent years. These skills could capture non-linearity and non-stationarity related to hydrological applications. Thus, data-driven methods based on AI have gained more attention for rainfall-runoff simulation [14].

In the last two decades, AI has been widely used for efficient simulating of nonlinear systems and capturing noise complexity in the datasets. For example, ANN and fuzzy logic are two popular AI-based approaches in flood prediction. Comparing with the classical black box models such as Auto Regressive (AR), Moving Average (MA), Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA), Auto Regressive Integrated Moving Average with exogenous input (ARIMAX), Linear Regression (LR), and Multiple Linear Regression (MLR) which are linear, AI-based models are nonlinear models which are able to capture non-stationarity and non-linearity features. As a result, more and more researchers have developed models that are able to overcoming the drawbacks of conventional models [15].

In the above, conventional machine learning techniques only have the ability to process natural data in their raw form without other insight information. However, Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. It could discover intricate structure in the data sets and change its internal parameters by using the backpropagation algorithms. Two of the most hot research points in deep learning are enhancing computer vision using CNN and modelling sequential data through RNN [16,17].

With conventional machine learning methods, we must extract features from data that are strongly correlated with dependent variables like ANN, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) etc. Deep learning could automatically extract features via the hidden layers. The hydrological process is always a kind of typical time sequential data. The traditional time-series simulation and prediction mainly rely on memoryless models [18] such as ANN and autoregressive (AG) models, they predict the next step in a time-series from a fixed number of previous steps. The RNNs can be trained to learn sequential or time-varying patterns by facilitate time delay units through feedback connections. The RNNs is particularly suitable for hydrological prediction in the context of giving a precise and timely prediction of time-series in the systems.

More modern RNN architectures were proposed since the late 1990s and one of the most successful RNN architectures is the Long Short-Term Memory (LSTM). This architecture has memory cells replaced the traditional hidden layer mode. The memory cells could store, write and read data via gates that open and close. These memory cells just like data in computer memory. LSTM is a dynamic model that has been used to simulate and predict sequences as music, text and motion capture data [19]. Besides, LSTM can be trained for sequence generation by processing real data sequences one step at a time and predicting what comes next.

However, to our knowledge, there are not so many studies using deep learning in hydrology, especially for large time-series datasets. Zhang [19] used LSTM networks to enhance internet of things for combined sewer overflow monitoring. Through a comparison of MLP, Wavelet Neural Network (WNN), LSTM and Gated Recurrent Unit (GRU), the LSTM and GRU had better performance for

multi-step-ahead time-series prediction. The same result was also gotten in the managing sewer in-line storage control using hydraulic model and recurrent neural network. The LSTM exhibits the superior capability for time-series prediction [19]. Kratzert [20] modeled rainfall-runoff with LSTM network. He found that LSTM could learn long-term dependencies between the provided inputs and outputs of the network. Using this approach, they achieved better model performance, which underlined the potential of the LSTM for hydrological modeling applications. The same conclusion also found in Fischer [17] making prediction of financial market using LSTM. He found the LSTM networks to outperform memory-free classification methods, i.e., A Random Forest (RAF), Deep Neural Net (DNN), and Logistic Regression Classifier (LRC). Thus, the LSTM network maybe a better choice for rainfall-runoff prediction.

In north-western China, there are complicated and changeable rainfall-runoff relationships [21]. The climate undergoes big changes in these years and underlying surface are changing with the development of China Society. Therefore, the prediction of runoff series in such regions should preferably be based on the existing long data with the memory networks. This novelty memory neural networks could better model the rainfall-runoff process and make accurate prediction. These methods possess human-like expertise with a specific domain adapt themselves and learn to do better in changing environments. Thus, it is a new try to use LSTM network to predict runoff and it is suitable for this changeable situation.

The objective of this study is to build real-time data-driven models that enable to simulate and predict rainfall-runoff from available data. This data-driven modelling analyzes relations between precipitation and runoff time-series. In this study, we selected 98 flooding events from 1971 to 2013 in Jingle hydrology station catchment basin. We use two types of neural network, namely ANN and LSTM. Although the machine learning algorithms such as RNNs provide real-time forecasting, it cannot give us an insight of the rainfall-runoff process. Besides, there are rare applications of LSTM in flood forecasting, as state of the art RNN architecture, the effectiveness of LSTM needs to be investigated. In this study, we hypothesized that the AI-based models have better performance in prediction rainfall-runoff and the modeling results in new architecture artificial neural network of LSTM may outperform ANN.

## 2. Methods

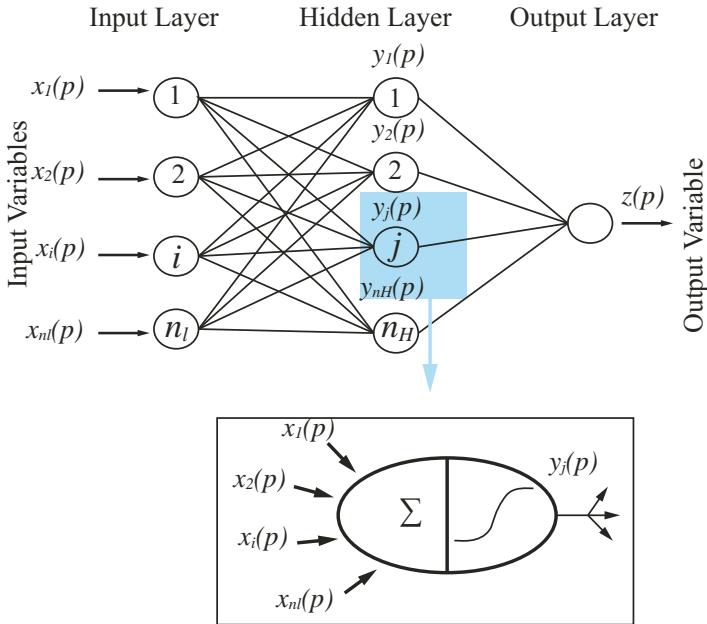
### 2.1. Artificial Neural Network

The ANN functions are similar to the human brain and nervous system which are a form of AI. ANNs can be trained with datasets to conduct prediction models and learn the intrinsic relationships without parameters [22]. These ANN models are being used as an efficient tool to reveal nonlinear relationship between inputs and outputs. Unlike conceptual models, using ANN models only dealing with mathematical relationship between inputs and outputs which are not defined. The commonly used ANN model (feed forward neural network) comprises of three layers of input, hidden and output (Figure 1). Each layer possesses a set of nodes (neurons) in which they are fully connected with nodes in the following layer. The model has a feed forward phase in which inputs signals propagate in forward direction (layer by layer) to reach output layer and an error backward propagation process which modifies the connection strengths (weights). Error is defined as the difference between computed and observed values of the target variable. Generally, the ANN model can be mathematically formulated as:

$$O_k = g_2 \left[ \sum_{j=1}^M W_{kj} g_1 \left( \sum_{i=1}^N W_{ji} x_i + W_{j0} \right) + W_{ko} \right] \quad (1)$$

where  $x_i$  is the input value to node  $i$ ,  $O_k$  is the output at node  $k$ ,  $g_1$  is activation function (nonlinear) for the hidden layer and  $g_2$  is activation function (linear) for the output layer.  $N$  and  $M$  represent

the number of neurons in the input and hidden layers, respectively.  $W_{j0}$  and  $W_{k0}$  are biases of the  $j$ th neuron in the hidden layer and the  $k$ th neuron in the output layer.  $W_{ji}$  is the weight between the input node  $i$  and the hidden node  $j$ , and  $W_{kj}$  the weight between the hidden node  $j$  and the output node  $k$ .



**Figure 1.** ANN architecture with one hidden layer (typical three-layer feed forward artificial neural networks) [10].

## 2.2. RNN

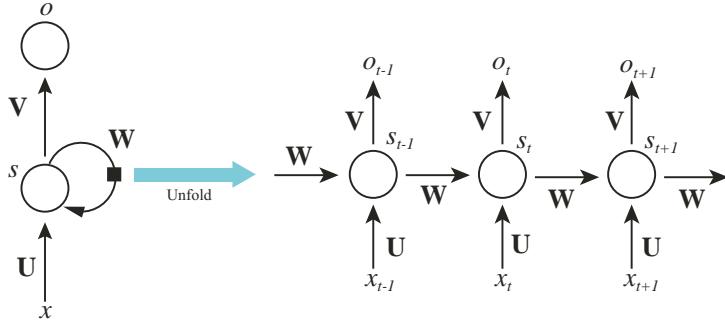
Recurrent neural networks (RNNs) are powerful model for sequential data. Recurrent neural network are a strict superset of feedforward neural networks, augmented by the inclusion of recurrent edges that span adjacent time steps, introducing a notion of time to the model [19]. While RNNs may not contain cycles among the conventional edges, recurrent edges may form cycles, including self-connections. At time  $t$ , nodes receiving input along recurrent edges receive input activation from the current example  $x^t$  and also from hidden nodes  $h^{t-1}$  in the network's previous state. The output  $\hat{y}^t$  is calculated given the hidden state  $h^t$  at that time step. Thus, input  $x^{t-1}$  at time  $t - 1$  can influence the output  $\hat{y}^t$  at time  $t$  by way of these recurrent connections (Figure 2).

We can show in two equations that all calculations are necessary for computation at each time step on the forward pass in a simple recurrent neural network:

$$h^{(t)} = \sigma(W_{hx}x + W_{hh}h^{(t-1)} + b_h) \quad (2)$$

$$\hat{y}^{(t)} = \text{softmax}(W_{gh}h^{(t)} + b_y) \quad (3)$$

where  $W_{hx}$  is the matrix of weights between the input and hidden layers and  $W_{hh}$  is the matrix of recurrent weights between the hidden layers at adjacent time steps. The vectors  $b_h$  and  $b_y$  are biases which allow each node to learn an offset.

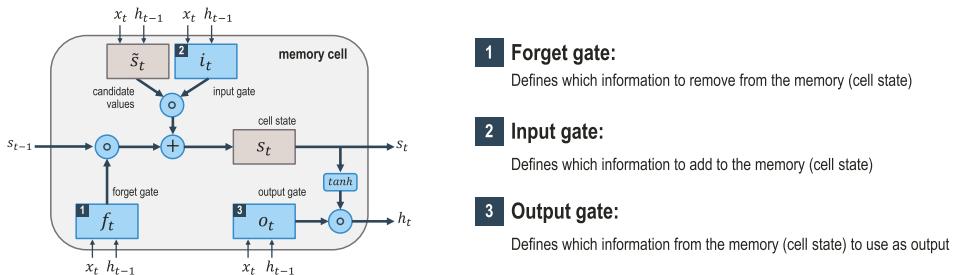


**Figure 2.** A simple RNN architecture with one hidden layer (recurrence using the previous hidden state).  $\mathbf{W}$ ,  $\mathbf{U}$ ,  $\mathbf{V}$  are parameters for weights [23].

### 2.3. LSTM

LSTM networks belong to the class of recurrent neural networks (RNNs), i.e., neural networks whose “underlying topology of inter-neuronal connections contains at least one cycle”. They have been introduced by Hochreiter and Schmidhuber [24] and were further refined in the following years. LSTM networks are specifically designed to learn long-term dependencies and are capable of overcoming the previously inherent problems of RNNs, i.e., vanishing and exploding gradients (Figure 3).

LSTM networks are composed of an input layer, one or more memory cells, and an output layer. The number of neurons in the input layer is equal to the number of explanatory variables. The main characteristic of LSTM networks is contained in the hidden layer consisting of so called memory cells. Each of the memory cells has three gates maintaining and adjusting its cell state  $s_t$ : a forget gate ( $f_t$ ), an input gate ( $i_t$ ), and an output gate ( $o_t$ ).



**Figure 3.** The architecture of LSTM memory block [17].

At every time-step  $t$ , each of the three gates is presented with the input  $x_t$  (one element of the ) as well as the output  $h_{t-1}$  of the memory cells at the previous time-step  $t - 1$ . Hereby, the gates act as filters, each fulfilling a different purpose:

- The forget gate defines what information is removed from the cell state.
- The input gate specifies what information is added to the cell state.
- The output gate specifies what information from the cell state is used

The sequential update formula are

Input node

$$g^{(t)} = \tanh(W_{gx}x^{(t)} + W_{gh}h^{(t-1)} + b_g) \quad (4)$$

Input gate

$$i^{(t)} = \sigma(W_{ix}x^{(t)} + W_{ih}h^{(t-1)} + b_i) \quad (5)$$

Forget gate

$$f^{(t)} = \sigma(W_{fx}x^{(t)} + W_{fh}h^{(t-1)} + b_f) \quad (6)$$

Output gate

$$o^{(t)} = \sigma(W_{ox}x^{(t)} + W_{oh}h^{(t-1)} + b_o) \quad (7)$$

Cell state

$$s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot o^{(t)} \quad (8)$$

Hidden gate

$$h^{(t)} = \tanh(s^{(t)}) \odot o^{(t)} \quad (9)$$

Output layer

$$y^{(t)} = (W_{hy}h^{(t)} + b_y) \quad (10)$$

where  $\sigma$  is the sigmoidal function,  $\odot$  is element wise multiplication,  $x^{(t)}$  is the input vector (forcings and static attributes) for the time step  $t$ ,  $W_s$  are the network weights,  $b_s$  are bias parameters,  $y$  is the output to be compared to observations,  $h$  is the hidden state, and  $s$  is called the cell state of memory cells, which is unique to LSTM.

#### 2.4. Performance Evaluation Criteria

In this study, performance of different models is assessed by statistical error measures and characteristic of flood process error including the coefficient of determination ( $R^2$ ), root mean square error ( $RMSE$ ), Nash-Sutcliffe Efficiency ( $NSE$ ), mean absolute error ( $MAE$ ), error of time to peak discharge ( $ET_p$ ) and error of peak discharge ( $EQ_p$ ).

$$R^2 = \frac{(\sum_{i=1}^n (y_i - \bar{y})(y'_i - \bar{y}'))^2}{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (y'_i - \bar{y}')^2} \quad (11)$$

where  $y'_i$  ( $m^3/s$ ) and  $y_i$  ( $m^3/s$ ) represent the discharge of the simulated and observed hydrographs at the time  $i$ ,  $\bar{y}$  ( $m^3/s$ ) and  $\bar{y}'$  ( $m^3/s$ ) denote the average observed and simulated discharge at the time  $i$  and  $n$  is the data points number. The coefficient of determination,  $R^2$ , known as the square of the sample correlation coefficient, ranges from 0 to 1 and describes the amount of observed variance explained by the model. A value of 0 implies no correlation, while a value of 1 suggests that the model can explain all of the observed variance.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y'_i)^2}{n}} \quad (12)$$

The root mean square error,  $RMSE$ , evaluates how closely that predictions match to observations, Values may range from 0 (perfect fit) to  $+\infty$  (no fit) based on the relative range of the data.

$$NSE = 1 - \frac{\sum_{i=1}^n (y_i - y'_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (13)$$

The Nash-Sutcliffe Efficiency,  $NSE$ , measures the model's ability to predict variables different from the mean and gives the proportion of the initial variance accounted for by the model. Where

NSE ranges from 1 (perfect fit) to  $-\infty$ . Values less than zero indicate that the observation mean would be a better predictor than the model.

$$MAE = \frac{\sum_{i=1}^n |y'_i - y_i|}{n} \quad (14)$$

The mean absolute error,  $MAE$ , measures the difference between observed and modelled results. It is an average of the absolute errors, where  $y'_i$  is the simulation and  $y_i$  is the observation.

$$ET_p = |T_{m,p} - T_{o,p}| \quad (15)$$

The error of time to peak discharge,  $ET_p$ , measures the model's time accuracy of peak runoff discharge prediction. Where  $T_{m,p}$  (hour) and  $T_{o,p}$  (hour) are the peak time for the modelled and observed peak runoff discharge, respectively.

$$EQ_p = \frac{(y_{m,p} - y_{o,p})}{y_{o,p}} \times 100\% \quad (16)$$

The error of peak discharge,  $EQ_p$ , measures the model volume accuracy of peak runoff discharge prediction. Where  $y_{m,p}$  ( $m^3/s$ ) and  $y_{o,p}$  ( $m^3/s$ ) are the modelled and observed peak runoff discharges, respectively.

## 2.5. The Approach and Modelling Process

In this study, data preparation and handing is entirely conducted in Python 3.5, relying on the packages numpy and pandas. Our LSTM and ANN networks are developed with keras on top of Google TensorFlow, a powerful library for large-scale machine learning on heterogenous systems.

We define the LSTM with 50 neurons in the first hidden layer and 1 neuron in the out layer for predicting discharge. The input shape will be 1 time step with 16 features. We will use the Mean Absolute Error (MAE) loss function and the efficient Adam version of stochastic gradient descent [25].

The type of ANN used in this study is a multi-layer-feed-forward perceptron (MLP) trained with the use of back propagation learning algorithm. The MLP network consists of input layer, hidden layer, and output layer. The final connection weights are kept fixed at the completion of training and new input patterns are presented to the network to produce the corresponding output consistent with the internal representation of the input/output mapping. In this study, the Levenberg–Marquardt (LM) algorithm is used for training the MLP network. The LM algorithm is often the fastest back propagation algorithm, and has been highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms. Further information on the back propagation learning algorithms can also be found in Dawson [26].

The simulation function of discharge is shown as:

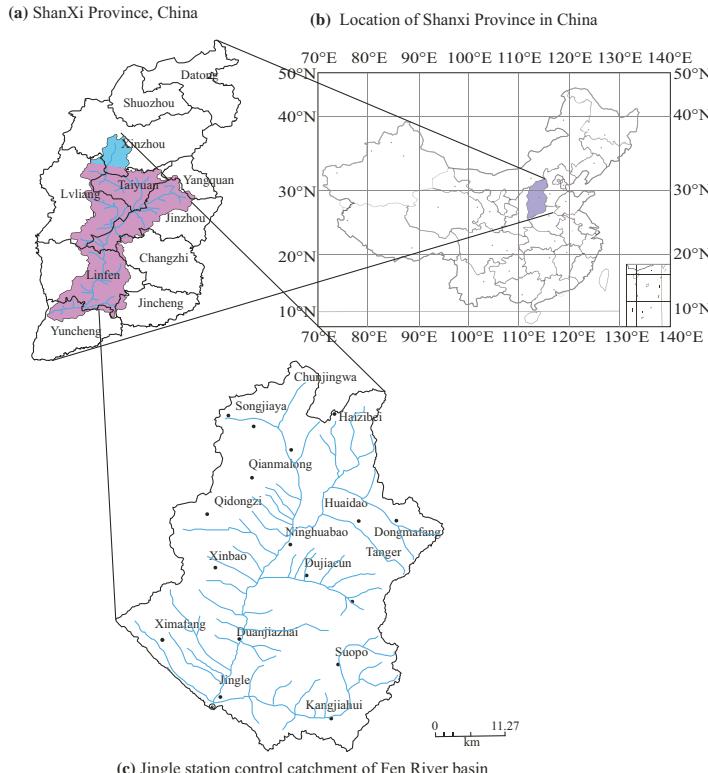
$$Q_t = f(Q_{t-n}, R_{t-n}, X_{t-n}) \quad (17)$$

in which  $Q_t$  is current flow,  $Q_{t-n}$  is antecedent flow (at  $t-1, t-2, \dots, t-n$  time steps),  $R_{t-n}$  is antecedent rainfall (at  $t-1, t-2, \dots, t-n$ ), and  $X_{t-n}$  represents any other factors identified as affecting  $Q_t$  (e.g., year type, percentage impervious area, storm occurrence). In this paper, we used the 14 rainfall stations and antecedent flow to forecast the runoff. We have chosen different values of  $n$  with 1, 2, 3, 4, 5, 6 (hour) indicating 6 types of lead time.

## 3. Case-Study

Fen River Basin ( $35^{\circ}20' - 39^{\circ}00'$  N latitude,  $110^{\circ}30' - 113^{\circ}32'$  E longitude) is located in Shanxi Province, North China (Figure 4). The Fen River is one of the largest tributaries of the Yellow River in its middle reach, joining the Yellow River in Hejing County. The river basin is bounded by Taihang

Mountain to the east, and Lvliang Mountain to the west, which also form the boundary between Yellow River and Fen River. Located in the eastern Loess Plateau of China, the climate of the Fen River Basin is temperate and sub-humid, with mean annual precipitation of 450 mm. In this area, the landforms are usually capped by a thick layer of loess due to dust deposition during the Quaternary. The study region is the catchment of Jingle hydrology station. The Jingle station was constructed in April on 1943 which was control station in main upstream of Fen River. The area of Jingle station controled basin is 2799 km<sup>2</sup> and the length of main stream is 83.9 km with average slope 0.67%. There are four tributaries in this basin, namely Hong river, Mingcun river, Dongnian river and Xinian river.



**Figure 4.** Location of the study site and the gauge stations. (a) Description of Fen River basin in Shanxi Province of China; (b) Description of Shanxi Province in Chian; (c) The control catchment of Jingle Station in Fen River and distribution of rainfall gauge stations.

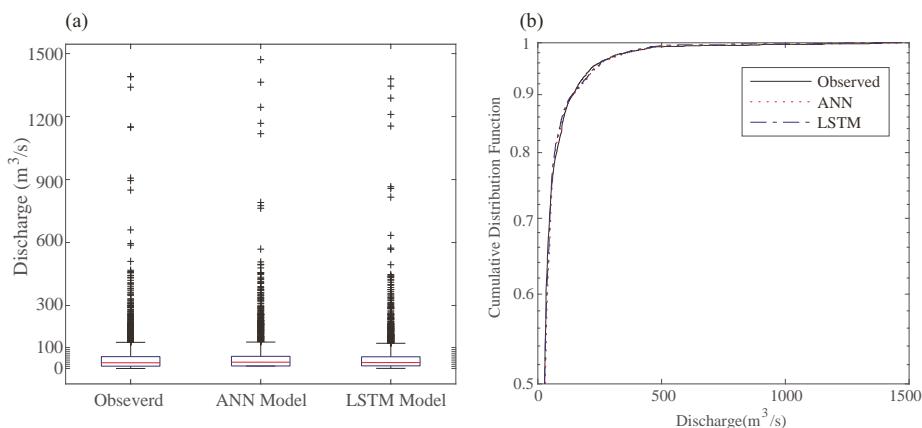
The annual mean precipitation in Jingle control basin is about 538.38 mm, the amount of mean flood in 24 h is about 50–55 mm, the maximum rainfall in the single site over 24 h is 109.6 mm. The average peak runoff and maximum peak runoff is 594 and 2230 m<sup>3</sup>/s. The rainfall station is shown by Figure 4c. The downstream Jingle discharge station is the forecasting object. This study collected hourly discharge data from Jingle station and hourly rainfall data from fourteen gauges. Data for 98 flood events from 1971 to 2013 with complete records were obtained. Among these flood events, 82 events (4962 datasets) were used for calibration and 12 events (1488 datasets) were used for validation red In this paper, we have chosen the typical rainfall-runoff process for validation to make the network models more representative, namely, the big volume discharge, the normal volume discharge in different periods from 1971 to 2013.

#### 4. Results

Every flood event is so different with rainfall duration, peak discharge, rainfall center (Table 1) that the process rainfall-runoff is difficult to learn. The Figure 5 illustrates that the statistical characteristics of 12 flood events data for validation. The upper boundary of Figure 5 is not above 150 ( $\text{m}^3/\text{s}$ ). The rapid flooding with large volume discharge in a short time makes many outliers in the dataset, but this typical large flooding is not common only 6 events (6.1%) (peak discharge above 1000 ( $\text{m}^3/\text{s}$ )) over the period from 1971 to 2013. Thus, we also seriously considered the sudden bigger data in constructing models. From Figure 5a, ANN model made some bigger forecasting values comparing with observed data when discharge data was exceeded 1200 ( $\text{m}^3/\text{s}$ ). While the LSTM model is better than ANN model at the same situation. The Figure 5b is shown the cumulative distribution of observed and modelled data. The three Lines almost coincided indicating that ANN and LSTM models have similar forecasting preferences in low volume discharge simulation. It also illustrates that the value of discharge among 0–200 ( $\text{m}^3/\text{s}$ ) takes percentage of almost 90%. From the analysis of dataset characteristics, we could find it is difficult for rainfall-runoff simulation taking into account sudden big and small volume of discharge. However, the above results lead to preliminary conclusion that ANN and LSTM models have better performances in flooding forecasting.

**Table 1.** Characteristics of collected flood events in Jingle discharge station.

Event No.	Date	Total Rainfall (mm)	Rainfall Duration (h)	Rainfall Center	Peak Discharge ( $\text{m}^3/\text{s}$ )
1	1 July 1971	8.86	36	Ninghuabao	164.50
2	23 July 1971	63.40	69	Chunjingwa	261.21
3	31 July 1971	10.44	12	Dongzhai	286.00
4	7 August 1971	21.07	42	Ninghuabao	184.14
5	15 August 1971	7.60	16	Chunjingwa	145.00
6	27 August 1971	15.71	36	Chunjingwa	112.00
7	31 July 1972	11.98	15	Huaidao	142.43
	...	...	...	...	...
92	10 October 2007	43.88	57	Chashang	106.00
93	23 September 2008	70.49	88	Qidongzi	132.00
94	10 August 2010	70.50	24	Songjinya	67.00
95	11 July 2011	41.88	24	Dujiacun	54.35
96	26 July 2012	40.57	41	Ninghuabao	134.00
97	30 July 2012	41.95	41	Chashang	61.90
98	17 July 2013	29.91	32	Jingle	74.40



**Figure 5.** Box-plots (a) and cumulative distribution function (b) of observed and estimated discharge for the 12 flood events of validation using ANN and LSTM models.

In the above study, we discussed statistical features of validation data. Then, the estimated hydrograph was used to compare performance of different models in validation (Figure 6). Even though the flooding process is difficult for simulation, the ANN and LSTM models all simulated well in general. Comparing with the peak discharge simulation, the value of ANN modelled were always bigger than observed data. In the low value of discharge simulation, the ANN modeled values appeared abnormal fluctuations. From estimated hydrograph Figure 6, it shows that LSTM model is more stable and simulated very well than ANN model. Thus, the LSTM model has better ability in nonlinearity simulation. Table 2 makes comparison of performances of ANN and LSTM models for runoff prediction. This is quantitative analysis of ANN and LSTM models using 6 preference criteria. The values of  $R^2$  and  $NSE$  are all beyond 0.95 in the LSTM modelling results in calibration and validation periods. Comparing with ANN model, the LSTM model values of  $RMSE$ ,  $MAE$ ,  $ET_p$  and  $EQ_p$  are all less than ANN indicating better performances in rainfall-runoff simulation. Especially, the ANN values of  $EQ_p$  are almost 4 times bigger than LSTM model. These cases illustrate that the LSTM model have accurately simulated peak discharge. The prediction of peak discharge of flood is critical for hydrological process simulation. Thus, the new model of LSTM with complicated architecture is a good choice for rainfall-runoff simulation and flood forecasting.

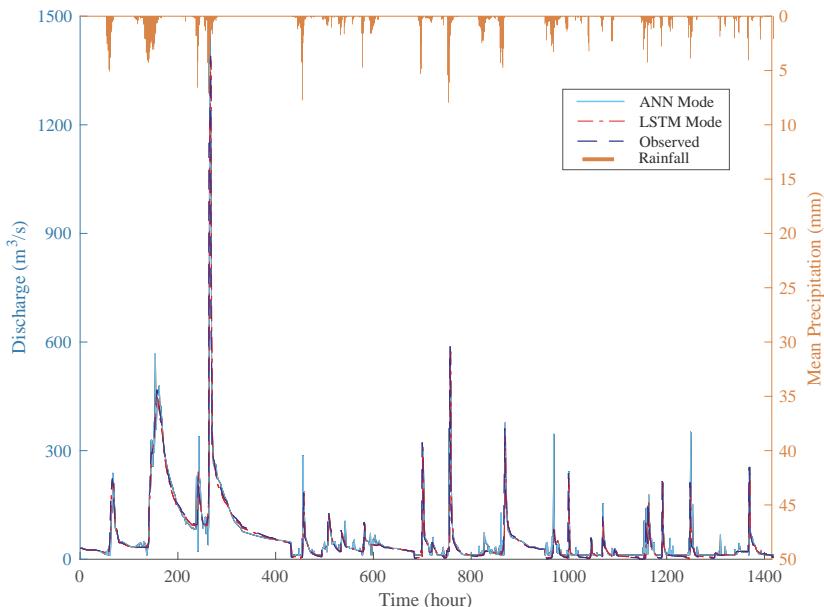
**Table 2.** Comparison of performances of ANN and LSTM models for runoff prediction (lead time = 1 h) at calibration (86 flood events) and validation (12 flood events) periods.

Events	Modes	$R^2$	$RMSE$ ( $m^3 s^{-1}$ )	$NSE$	$MAE$ ( $m^3 s^{-1}$ )	$ET_p$ (h)	$EQ_p$
Calibration							
86 events series	ANN	0.81	124.21	0.83	47.23	5.4	12%
	LSTM	0.95	45.12	0.97	12.4	2.6	4%
Validation							
12 events series	ANN	0.83	35.6	0.83	23.6	3.7	14%
	LSTM	0.96	12.4	0.96	6.3	1.4	3%

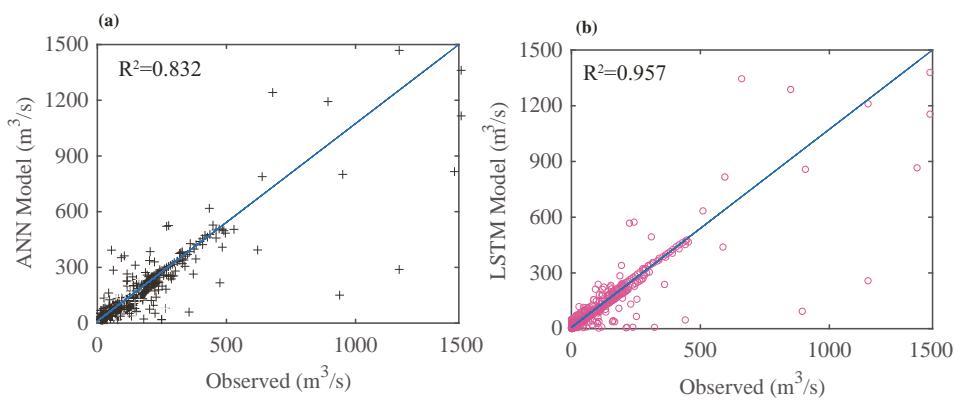
After quantitative and qualitative analysis of ANN and LSTM models, we also scatter the observed and simulated discharge values (Figure 7). The values of ANN and LSTM models'  $R^2$  are 0.832 and 0.957, respectively. The LSTM model has higher values of  $R^2$  indicating that this model could well reflect the relationship between observed and simulated discharge. From the Figure 7a, the data is scattered more loose in ANN model, while it is relatively closer to the line in LSTM model (Figure 7b). It is clearly shown that LSTM model is better than ANN model in runoff prediction which has better correlation with observed data. Besides, the values are almost near the fit line in the two models. However, the two models all appear some abnormal values. The reason of this phenomenon is that ANN and LSTM models have some fluctuations under the suddenly changes in rainfall and discharge data.

We have talked about general characteristics of ANN and LSTM models in above study. However, some special features need to deeply insight into ANN and LSTM models for hydrological process simulation. The Figure 8 shows observed and estimated hydrographs of the ANN and LSTM models at the validation stage in 12 flood events. Among the 12 flooding events, only peak discharge of event 2 was beyond 1000 ( $m^3/s$ ). The ANN model has bed ability of peak discharge prediction comparing with LSTM model. In the flooding event 1, 3, 4, 8, 9, and 11, the simulated peak discharge always higher than observed. These modelled values of peak discharge were not to be trusted in flooding event 7, 8 and 11 with abnormally bigger values. However, the LSTM model was proved more reliable in prediction of peak discharge. We can take flooding event 4, 7, 9 and 10 that ANN model always has much sensitivity to rainfall. The simulated values of ANN model fluctuate abnormally comparing with observed values no matter big or small volume discharge. While the LSTM model don't appear these performances. The differences in ANN and LSTM model architectures are memory cells. The various

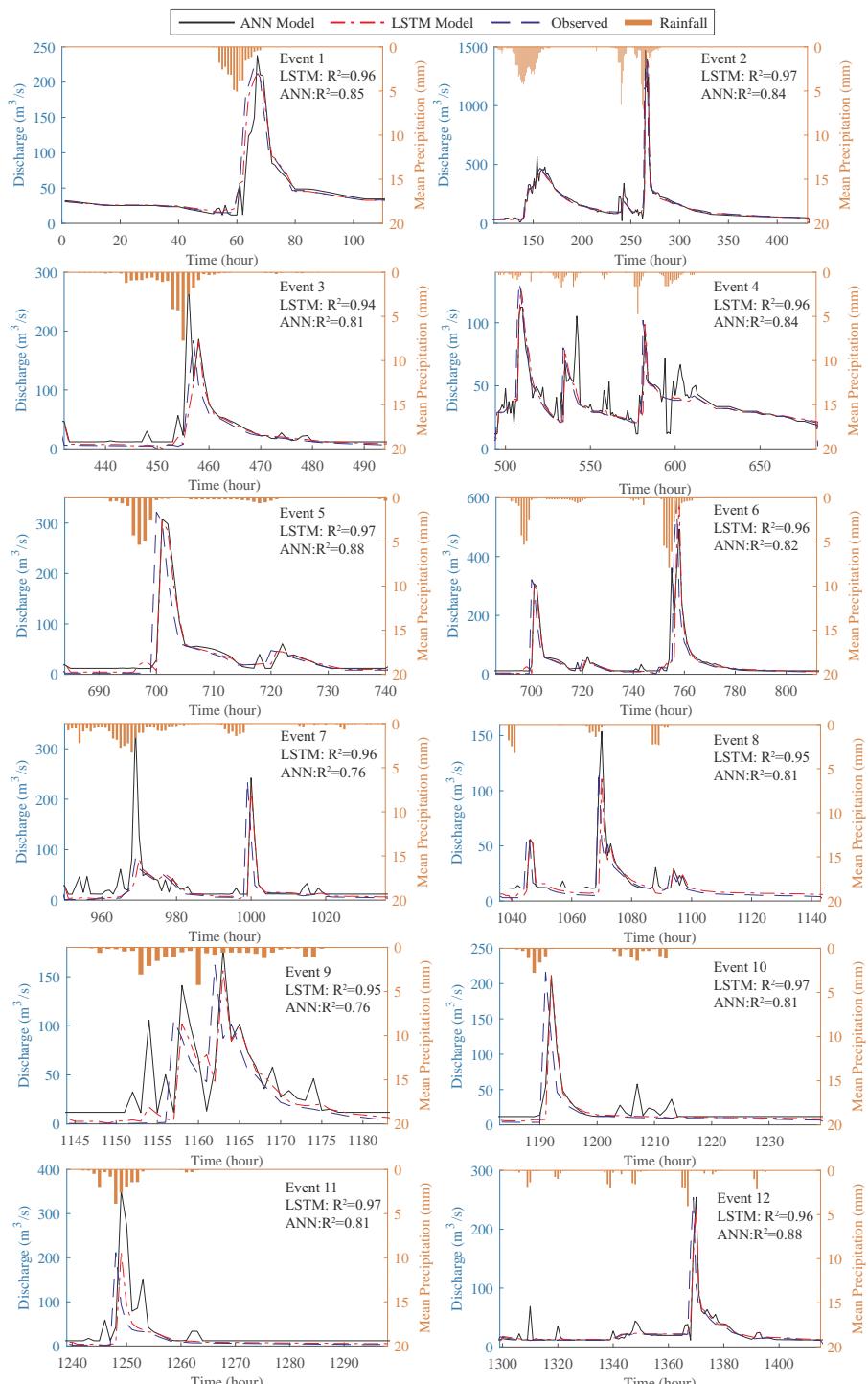
memory cells have ability to filter data and memory data features making as deep learning function to simulate rainfall-runoff process. The disadvantages of ANN model are obvious. Compared with ANN and LSTM models in these flooding event simulation, it is proved that LSTM model is more intelligence than ANN model in predicting rainfall-runoff.



**Figure 6.** The observed and estimated hydrographs (12 flood events of validation) using ANN and LSTM models.



**Figure 7.** Scatter plot of the observed and the simulated runoff during 12 validation flood events.  
 (a) ANN model; (b) LSTM model.



**Figure 8.** Observed and estimated hydrographs of the ANN and LSTM model at the validation stage in 12 flood events.

We have already discussed ANN and LSTM models simulation performances using lead time 1 h in above. The Table 3 illustrates runoff forecasting at different lead times (1–6 h) by ANN and LSTM model. In general, LSTM model had better simulation results than ANN model at different lead times. In the calibration and validation stage, the values of performances criteria in LSTM model are all better than ANN models. Comparing with different lead time situations, the values of  $R^2$  and NSE were reducing with the increasing lead time. The values of RMSE, MAE and  $ET_p$  did not show clearly changing law. The LSTM values of  $EQ_p$  was the smallest in lead time at 1 h. While the ANN model had badly performances in lead time at 6 h as the values of  $R^2$  and NSE near 0.7. Even though LSTM prediction ability was inducing with large lead time, the values of  $R^2$  and NSE still above 0.8. Compared with the ANN model, LSTM also has the low value of  $ET_p$  and  $EQ_p$ . These results illustrate that the LSTM has the better performances in forecasting peak discharge in each flood event. These results mean that the chosen of LSTM model is suitable for the rainfall-runoff modeling. From all of these results, we can considerer LSTM network suitable using in hydrology research.

**Table 3.** The performances of runoff forecasting at different lead times (1–6 h) by ANN and LSTM model for series flood events.

Lead Time (h)	Data	Models	$R^2$	RMSE ( $\text{m}^3 \text{s}^{-1}$ )	NSE	MAE ( $\text{m}^3 \text{s}^{-1}$ )	$ET_p$ (h)	$EQ_p$
1	Calibration	ANN	0.81	124.21	0.83	47.23	5.4	12%
		LSTM	0.95	45.12	0.97	12.4	2.6	4%
	Validation	ANN	0.83	35.6	0.83	23.6	3.7	14%
		LSTM	0.96	12.4	0.96	6.3	1.4	3%
	Calibration	ANN	0.83	132.2	0.86	42.13	11.4	13%
		LSTM	0.95	42.12	0.94	13.4	2.4	7%
2	Validation	ANN	0.79	23.6	0.85	23.1	2.7	12%
		LSTM	0.93	15.4	0.95	6.3	1.8	13%
	Calibration	ANN	0.78	164.21	0.79	56.23	14.4	11%
		LSTM	0.91	47.12	0.91	13.4	2.8	6%
	Validation	ANN	0.81	25.6	0.78	23.6	4.2	15%
		LSTM	0.92	14.4	0.91	7.3	1.4	16%
3	Calibration	ANN	0.81	144.21	0.82	48.23	11.4	12%
		LSTM	0.91	65.12	0.91	15.4	2.8	12%
	Validation	ANN	0.72	37.8	0.81	25.6	3.1	11%
		LSTM	0.91	13.4	0.93	11.3	1.6	15%
	Calibration	ANN	0.78	135.21	0.81	48.23	11.4	12%
		LSTM	0.87	49.12	0.81	17.4	4.6	8%
4	Validation	ANN	0.74	38.6	0.79	24.6	5.7	16%
		LSTM	0.84	22.4	0.91	6.3	1.4	17%
	Calibration	ANN	0.71	144.21	0.73	67.23	18.4	17%
		LSTM	0.84	48.12	0.96	13.4	2.7	12%
	Validation	ANN	0.75	25.6	0.79	23.6	3.7	14%
		LSTM	0.83	14.4	0.85	8.3	2.4	18%

## 5. Discussion and Conclusions

The process of rainfall-runoff simulation is critical for hydrology [27]. However, the process of rainfall-runoff is a complex problem for the hydrological modelling. Saturated and infiltration excess runoff could all appear in one rainfall-runoff event in semi-dry and semi-humid region. Conducting suitable models is more complicated into semi-dry and semi-humid regions. The mechanism of runoff generation is more complicated than humid region. Considering the features of climate and hydrological process, lots of watershed belong to semi-dry and semi-humid region in China. Thus, the performances of physical models and conceptual models were badly used that the correlation coefficients were around 0.6 in this semi-dry and semi-humid regions [28,29]. However, in a recently study of rainfall-runoff simulation, various artificial networks were used for the simulation and prediction [30]. In this study, we use the traditional network as ANN and the new deep learning

network as LSTM for the simulation. In generally, LSTM model is better than the traditional ANN model. Because of the typical flood characteristics, the ANN models can not make accurate simulation [31], but the ANN models are still better than the physical models in this region. It is the progress of the AI based techniques making the revolutionary strides for hydrology [4].

Compared with other network models, Kan [31] used a hybrid data-driven (network model and physical model) models for event-based rainfall-runoff simulation. PEK model (hybrid model) outperformed other models with values of  $NSE$  and  $R^2$  are 0.51 and 0.73, respectively in validation stage. However, the results of this study all better than Kan's. There are two factors as inputs and model architecture that affect results of model outputs. In this paper, we used 14 rainfall stations data and antecedent discharge as the inputs. The dataset in this paper was larger than Kan's. We used the network model with memory cells (LSTM) that was progressed than his model. Thus, we got the better simulation performances. Lin [30] forecasted the typhoon-rainfall with a hybrid neural network model (the Self-Organizing Map (SOM) and Multilayer Perceptron Network (MLPN)). In Lin's study, SOM network was used for classification rainfall and then the MLPN was used for prediction. This model can forecast more precisely than the model developed by the conventional neural approaches, but the values of  $NSE$  were below 0.85. These values were also smaller than LSTM modeling results in this study. The reason was that LSTM model with memory cells could learn more from the datasets and accurately make simulation.

However, the hydrological cycle process significantly changed under human activities and climate changes in Loess Plateau where have implemented project of returning farmland to forest and protecting natural forecast from 1980s. The changeable environments also make influence on rainfall-runoff process [32]. It is important to test if the LSTM or ANN model could be used in this region. Compared with simulated 12 flooding events (lead time 1 h Figure 8), the values of correlation efficient were beyond 0.95 indicating that LSTM model was adaptable among different situations. Besides, the ANN model had bad adaptability with many abnormal simulations in changeable environment. In this study, LSTM model still had better performance when lead time was 6 h. Thus, LSTM could be used in this region for flooding prediction.

Compared with the previous study in rainfall-runoff modeling, the results of LSTM modeling have the higher values of  $R^2$  and  $NSE$ . The LSTM model had perfect performances in this paper, while it needed to be validated in numerous watersheds. Thus, the more and more studies need to study deep learning model (LSTM) application in hydrology. And finding the meaning of intrinsic structure parameters of LSTM can also improve our learning of hydrology process. Then, the AI techniques may accurately be applied in hydrology.

In this research, we used ANN and LSTM models for forecasting hourly runoff discharges in Jingle hydrology station control catchment basin. Comparing with conceptual and physical-based models, these black box models can well simulate rainfall-runoff process with excellent performance evaluation criteria. Compared with flooding events simulation, ANN model is more sensitive that has many abnormal fluctuations, while LSTM model is more intelligence than ANN model. In this study, the runoff is changed in time-series that the data is time related. The ANN model is constructed by fitting the different characteristics of the current state and making prediction. While LSTM model not only take full advantage of the current data characteristics but also use its gate structure to decide to remember or forget the previous features. With the progress of AI techniques, the deep learning methods of long short-term memory network could be better used in the hydrological simulation. The values of  $R^2$  and  $NSE$  in LSTM model are bigger than 0.9 when lead time is 1 h. With increment of lead time, values of performance criteria ( $R^2$  and  $NSE$ ) were slightly decreasing, but the values of LSTM model were still beyond 0.8 with good simulation abilities. It is because of LSTM is very effective in modeling time-series data, it can also be applied to weather forecasting, for example rainfall, fog and haze, stream flow et al. In this paper, we considered the data of the preceding hours to predict the runoff of the next hours. In the future, we could forecast different length of time or not only runoff forecasting, we can predict the entire sequence of data at the next moment. This deep learning

networks have better performances in hydrological time-series prediction. More researches will be needed in modelling hydrological process using deep machine learning.

**Author Contributions:** Conceptualization, C.H. and S.J.; Methodology, Z.L.; Software, H.L.; Validation, C.H., S.J. and Z.L.; Formal Analysis, Q.W.; Investigation, C.H.; Resources, C.H. and S.J.; Data Curation, N.L.; Writing—Original Draft Preparation, Q.W. All authors contributed to the final version of the manuscript.

**Funding:** This research received funding by [National Key Research Priorities Program of China] grant number [2016YFC0402402], [National Natural Science Foundation of China] grant number [61502434] and [China Postdoctoral Science Foundation] grant number [2017M620336].

**Acknowledgments:** We thank five anonymous reviewers for their valuable comments to improve the manuscript. We also gratefully thank Shan-e-hyder Soomro for his revising language problems. We also thank my other colleagues' valuable comments and suggestions that have helped improve the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Mlv, M.; Todini, E.; Libralon, A. A Bayesian decision approach to rainfall thresholds based flood warning. *Hydrol. Earth Syst. Sci. Discuss.* **2006**, *2*, 413–426. [[CrossRef](#)]
- Bartholmes, J.C.; Thielen, J.; Ramos, M.H.; Gentilini, S. The european flood alert system EFAS-Part 2: Statistical skill assessment of probabilistic and deterministic operational forecasts. *Hydrol. Earth Syst. Sci.* **2009**, *13*, 141–153. [[CrossRef](#)]
- Park, D.; Markus, M. Analysis of a changing hydrologic flood regime using the variable infiltration capacity model. *J. Hydrol.* **2014**, *515*, 267–280. [[CrossRef](#)]
- Meng, C.; Zhou, J.; Tayyab, M.; Zhu, S.; Zhang, H. Integrating artificial neural networks into the VIC Model for rainfall-runoff Modeling. *Water* **2016**, *8*, 407. [[CrossRef](#)]
- Lee, H.; McIntyre, N.; Wheater, H.; Young, A. Selection of conceptual models for regionalisation of the rainfall-runoff relationship. *J. Hydrol.* **2005**, *312*, 125–147. [[CrossRef](#)]
- Calver, A. Calibration, sensitivity and validation of a physically-based rainfall-runoff model. *J. Hydrol.* **1988**, *103*, 103–115. [[CrossRef](#)]
- Kan, G.; Li, J.; Zhang, X.; Ding, L.; He, X.; Liang, K.; Jiang, X.; Ren, M.; Li, H.; Wang, F.; et al. A new hybrid data-driven model for event-based rainfall-runoff simulation. *Neural Comput. Appl.* **2016**, *28*, 2519–2534. [[CrossRef](#)]
- Talei, A.; Chua, L.H.C.; Quek, C. A novel application of a neuro-fuzzy computational technique in event-based rainfall-runoff modeling. *Expert Syst. Appl.* **2010**, *37*, 7456–7468. [[CrossRef](#)]
- Taormina, R.; Chau, K.W. Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines. *J. Hydrol.* **2015**, *529*, 1617–1632. [[CrossRef](#)]
- Hsu, K.; Gupta, H.V.; Sorooshian, S. Artificial Neural Network Modeling of the Rainfall—Runoff Process. *Water Resour. Res.* **1995**, *31*, 2517–2530. [[CrossRef](#)]
- Radfar, A.; Rockaway, T.D. Captured runoff prediction model by permeable pavements using artificial neural networks. *J. Infrastruct. Syst.* **2016**, *22*, 04016007. [[CrossRef](#)]
- Salas, J.D.; Markus, M.; Tokar, A.S. Streamflow forecasting based on artificial neural networks. *Artif. Neural Netw. Hydrol.* **2000**, *36*, 23–51.
- Tokar, A.S.; Johnson, P.A. Rainfall-runoff modeling using artificial neural networks. *J. Hydrol. Eng.* **1999**, *4*, 232–239. [[CrossRef](#)]
- Chang, T.K.; Talei, A.; Alaghmand, S.; Ooi, M.P.L. Choice of rainfall inputs for event-based rainfall-runoff modeling in a catchment with multiple rainfall stations using data-driven techniques. *J. Hydrol.* **2017**, *545*, 100–108. [[CrossRef](#)]
- Yu, Y.; Zhang, H.; Singh, V. Forward prediction of runoff data in data-scarce basins with an improved ensemble empirical mode decomposition (EEMD) Model. *Water* **2018**, *10*, 388. [[CrossRef](#)]
- Chen, Y.; Fan, R.; Yang, X.; Wang, J.; Latif, A. Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning. *Water* **2018**, *10*, 585. [[CrossRef](#)]
- Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [[CrossRef](#)]

18. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
19. Zhang, D.; Martinez, N.; Lindholm, G.; Ratnaweera, H. Manage sewer In-Line storage control using hydraulic model and recurrent neural network. *Water Resour. Manag.* **2018**, *32*, 2079–2098. [[CrossRef](#)]
20. Kratzert, F.; Klotz, D.; Brenner, C.; Schulz, K.; Herrnegger, M. Rainfall-Runoff modelling using Long-Short-Term-Memory (LSTM) networks. *Hydrol. Earth Syst. Sci.* **2018**. [[CrossRef](#)]
21. Wan, R.; Shan, G. Progress in the hydrological impact and flood response of watershed land use and land cover change. *J. Lake Sci.* **2004**, *16*, 258–264.
22. Tayfur, G.; Singh, V.P. ANN and fuzzy logic models for simulating event-based Rainfall-Runoff. *J. Hydraul. Eng.* **2006**, *132*, 1321–1330. [[CrossRef](#)]
23. Zhang, D.; Lindholm, G.; Ratnaweera, H. Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. *J. Hydrol.* **2018**, *556*. [[CrossRef](#)]
24. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
25. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *Comput. Sci.* **2014**, arXiv:1412.6980.
26. Dawson, C.; Wilby, R. An artificial neural network approach to rainfall-runoff modelling. *Int. Assoc. Sci. Hydrol. Bull.* **1998**, *43*, 47–66. [[CrossRef](#)]
27. Jhong, Y.D.; Chen, C.S.; Lin, H.P.; Chen, S.T. Physical hybrid neural network model to forecast typhoon floods. *Water* **2018**, *10*, 632. [[CrossRef](#)]
28. Hu, C.; Guo, S.; Xiong, L.; Peng, D. A Modified Xinanjiang Model and Its Application in Northern China. *Hydrol. Res.* **2005**, *36*, 175–192. [[CrossRef](#)]
29. Li, Q. Analysis and discussion related to the hydrological watershed models used in the first hydrological forecasting technology competition of China. *Adv. Water Sci.* **1998**, *9*, 191–195.
30. Lin, G.F.; Jhong, B.C.; Chang, C.C. Development of an effective data-driven model for hourly typhoon rainfall forecasting. *J. Hydrol.* **2013**, *495*, 52–63. [[CrossRef](#)]
31. Kan, G.; Yao, C.; Li, Q.; Li, Z.; Yu, Z.; Liu, Z.; Ding, L.; He, X.; Liang, K. Improving event-based rainfall-runoff simulation using an ensemble artificial neural network based hybrid data-driven model. *Stoch. Environ. Res. Risk Assess.* **2015**, *10*, 1345–1370. [[CrossRef](#)]
32. Huang, G.; Rui, X.; Shi, P. Analysis of rainfall-runoff characteristics of Jing-Luo-Wei river basin. *Adv. Sci. Technol. Water Resour.* **2004**, *24*, 21–23.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Flood Routing Model with Particle Filter-Based Data Assimilation for Flash Flood Forecasting in the Micro-Model of Lower Yellow River, China

Minghong Chen <sup>\*</sup>, Juanjuan Pang and Pengxiang Wu

College of Water Resources and Civil Engineering, China Agricultural University, Beijing 100083, China; 18801061663@163.com (J.P.); 18800101270@163.com (P.W.)

\* Correspondence: chenminghong@cau.edu.cn; Tel.: +86-134-6666-4512

Received: 10 October 2018; Accepted: 5 November 2018; Published: 9 November 2018

**Abstract:** Reliable real-time flood forecasting is a challenging prerequisite for successful flood protection. This study developed a flood routing model combined with a particle filter-based assimilation model and a one-dimensional hydrodynamic model. This model was applied to an indoor micro-model, using the Lower Yellow River (LYR) as prototype. Real-time observations of the water level from the micro-model were used for data assimilation. The results show that, compared to the traditional hydrodynamic model, the assimilation model could effectively update water level, flow discharge, and roughness coefficient in real time, thus yielding improved results. The mean water levels of the particle posterior distribution are closer to the observed values than before assimilation, even when water levels change greatly. In addition, the calculation results for different lead times indicate that the root mean square error of the forecasting water level gradually increases with increasing lead time. This is because the roughness value changes greatly in response to unsteady water flow, and the incurring error accumulates with the predicted period. The results show that the assimilation model can simulate water level changes in the micro-model and provide both research method and technical support for real flood forecasting in the LYR.

**Keywords:** flood forecasting; data assimilation; particle filter algorithm; micro-model; Lower Yellow River

## 1. Introduction

Floods are among the most severe hazards in the world, causing loss of life and excessive damage to property. Thus, if floods can be reliably forecast, proper actions can be initiated to mitigate the damages during the flood event. The ability to provide reliable forecast of river water levels for a short period following a storm with significant precipitation is very important for both flood control and river regulation [1]. This study focused on the development of a flood routing model for real time flood forecasting.

In general, flood routing can be classified into two categories: the hydrologic method and the hydraulic method [2]. However, the hydrologic method for a hydrologic system cannot achieve the desirable degree of accuracy and lacks water level and detailed flow information [3]. In contrast, the hydraulic method can be widely applied to special problems, e.g., to forecast discharges and stages at cross-sections along rivers [4–7]. However, due to input uncertainty, model structure uncertainty, and parameter uncertainty in hydraulic models, flood forecast reliability tends to decrease with increased forecast lead period, so that the simulated results rarely fit the measurements. This is even worse in real-time flood forecasting. Hence, the hydraulic model needs to be improved to enable real-time flood forecasting.

The data assimilation algorithm continuously integrates real-time observational data into numerical models and adjusts both the motion state and parameters, which greatly improves the prediction accuracy of the models. Compared to other assimilation algorithms, the particle filter (PF) assimilation algorithm is suitable for both nonlinear and non-Gaussian systems. It can effectively estimate the uncertainties of the model and improve the prediction accuracy of the model simulation [8–10]. Han and Li [11] and Pan [12] compared the differences between the PF algorithm and the Kalman filter (KF) algorithm, and found that the PF algorithm can better solve the problem of a nonlinear non-Gaussian error in the model. In the nonlinear Lorenz system, the PF algorithm achieves better stability and accuracy than the KF algorithm [13–16]. Moradkhani et al. [17] used the PF algorithm to estimate the model parameters and state variables in the hydrological model, and found that the results fitted the observed values well. These studies showed that the PF algorithm is a very efficient method to solve problems of nonlinear systems such as flood routing systems. However, it is still rarely applied to real-time flood forecasting.

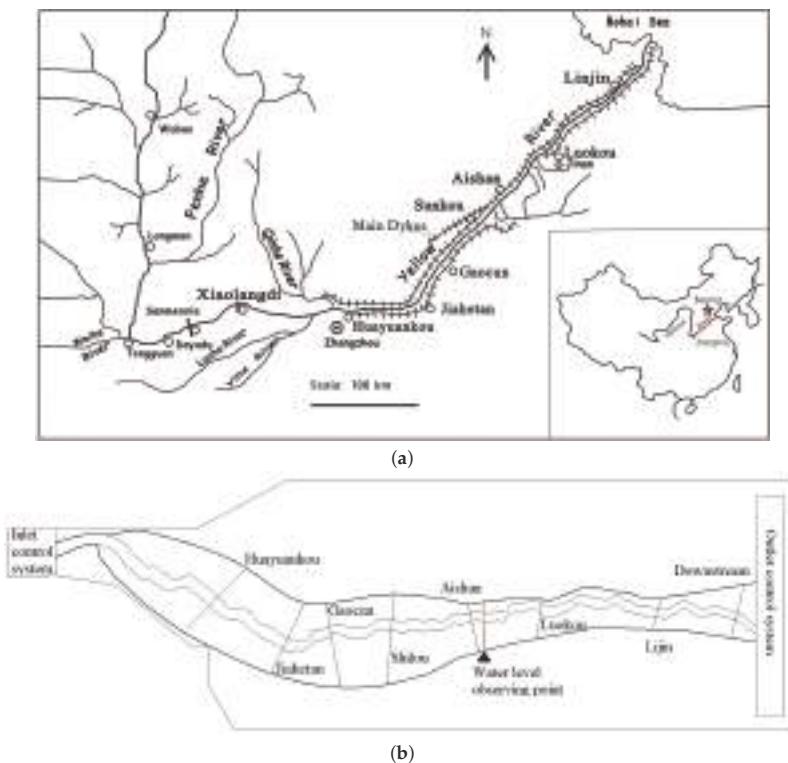
The purpose of this study was to verify and evaluate the accuracy of the assimilation model system for real-time flood prediction in a laboratory micro-model. Although this model can also be applied to river prototypes, due to the difficulty and cost for obtaining real-time hydrological data, a micro-model was established based on the Lower Yellow River (LYR) for flood experiments. An automatic flow control system and a real-time water level monitoring system were installed on the model to control the upstream flow discharge and the acquisition of real-time water level data of the observation points. The PF algorithm was applied to increase the quality of the predicted water levels. The practical motivation of this study was to improve the application of indoor data assimilation systems, and apply it to real-time forecasting of real floods in the LYR in future. This will improve forecasting accuracy, increase the lead time, and provide sufficient response time for flood control and disaster relief prior to the flood event.

## 2. Materials and Methods

### 2.1. Study Area and LYR Micro-Model

The study area is located in the lower reach of the Yellow River, in central China, stretching from the Xiaolangdi Reservoir to the Yellow River estuary, and comprising a distance of approximately 800 km (Figure 1a). The bed slope varies from 0.010% to 0.012%. The drainage area of the LYR is 23,000 km<sup>2</sup>. The width between the levees varies from 1 to 4 km; however, the width of the main channel is only about 500 m. Most of the cross-sections are compound sections. The wide flood plains between dykes are extensively cultivated. Here, more than 2000 natural villages had formed by 1900 and nearly 1.8 million people live on these flood plains. The total area of the flood plains is 3956 km<sup>2</sup>, accounting for 85% of the total area between the levees.

It is possible to use the real LYR as a study case for the verification of the validity of the assimilation model. However, it is expensive to establish a stable data acquisition and transmission system on such a long river. To verify the performance of the assimilation model under various hydrological conditions, an extended data acquisition duration is required. Thus, a micro-model was constructed using the LYR as a prototype in our laboratory to obtain real-time data more conveniently. This model was first designed in reference to digital elevation model (DEM) data of the LYR and then corrected by the cross-section data measured in 2013 before the flood season. The model has a total length of 27.38 m with a longitudinal ratio 1:28,000 and a total of 57 cross-sections along the river, including Huayankou, Jiahetai, Gaocun, Shilou, Aishan, Luokou, and Lijin (Figure 1b). Two automatic control systems are assumed for the model, one at the inlet that controls the inlet flow and one at the outlet that controls the downstream water level. There is also an automatic observation system at the water level observation point (see in Figure 1b), which obtains the water level in real time.



**Figure 1.** Map of the Lower Yellow River (LYR): (a) the location of the LYR; and (b) plan view of the micro-model in the laboratory.

## 2.2. Experimental Setup

All measuring equipment was calibrated prior to the experiment. Designed sequential processes of inlet flow discharge and outlet water levels were input into the controlling computer. Once the flood started, the water level observation system began to continually collect water level data at an interval of 10 s.

To verify the performance of the assimilation model under different hydrological conditions, several test events were designed in reference to the peak flow and flood duration in the prototype. Nine experiments were conducted for this study. The first three experiments constituted a calibration group which was used to calibrate the roughness coefficient of the hydrodynamic model only, while the further six experiments were used for the verification of the assimilation model. These six verification experiments were labeled Experiment 1–6 and were divided into two groups with three typical flow processes each. The first group was used to verify the prediction accuracy of the assimilation model for flood events without floodplain flooding, with one floodplain flooding, and with multi-floodplain flooding. The ranges of flow discharge were  $3.6\text{--}7.2 \text{ m}^3 \text{ h}^{-1}$ ,  $3\text{--}7.92 \text{ m}^3 \text{ h}^{-1}$  and  $7.2\text{--}11.88 \text{ m}^3 \text{ h}^{-1}$ , respectively. The second group was used to verify the prediction accuracy of the assimilation model during flood events with different numbers of flood peaks due to the periodically changing flood that may occur in the proto-river. All flow discharges in this group remained within  $7.2\text{--}16.2 \text{ m}^3 \text{ h}^{-1}$ .

### 2.3. Hydrodynamic Assimilation System

The hydrodynamic assimilation system mainly consists of a hydrodynamic mathematical model, an assimilation model, and a real-time data acquisition system. Details of the mathematical model principle and the model coupling method are described below.

#### 2.3.1. Hydrodynamic Mathematical Model

The one-dimensional hydrodynamic model has been widely used in flood forecasting due to its advantages of simple execution, short calculation time, and high real-time efficiency [18,19]; consequently, it was also used herein to meet the timeliness requirement of flood forecasting. The unsteady one-dimensional free surface flow in an open channel was described by the Saint-Venant equations [4]. The four-point implicit finite-difference scheme was used to discretize the governing equations. The equations were then solved by successively applying the double sweep Thomas algorithm under given boundary conditions and initial conditions [7]. The upstream conditions are inflow discharges and the downstream boundary conditions are outflow water levels.

The Manning coefficient of roughness reflects the roughness of the inner boundary of the channel and characterizes the resistance to water flow. The roughness value not only depends on the physical properties of the channel, but also on the state of the water flow. The overall roughness coefficient of a given cross-section is a composite value that includes the entire influence of water flow resistance. Typically, the roughness coefficient is determined by the measured hydrological observation data. Considering the change of channel characteristics along the river and the unsteadiness of the flooding flow, the spatiotemporal variation of the roughness coefficient should be considered in a flood forecasting hydrodynamic model. Therefore, it is the most important parameter that needs to be continually adjusted in the assimilation model.

#### 2.3.2. PF-Based Assimilation Model

The PF algorithm is a method that combines the Monte Carlo method with Bayesian estimation. It uses a set of random particles with correlated weights to approximate the probability density function and obtain the optimal probabilistic interval estimation. When the number of particles is sufficient, the posterior probability distribution of the state variables can approximate the true probability distribution [20,21]. Supposing a set of random weighted samples can be independently obtained from the posterior probability distribution of state variables, the posterior probability distribution of state variables can be expressed by the following formula:

$$p(X_t|X_{1:t}) \approx \sum_{i=1}^N w_{i,t} \delta(X_t - X_{i,t}) \quad (1)$$

where  $p(X_t|X_{1:t})$  represents the posterior probability distribution of state variables at time  $t$ ;  $X_{i,t}$  and  $w_{i,t}$  represent the  $i$ th particle and its weight, respectively, at time  $t$ ;  $X_t$  represents the data observation at time  $t$ ;  $N$  represents the number of particles; and  $\delta(\bullet)$  represents the Dirac delta function.

The water level, flow discharge, and roughness coefficient of each cross-section along the river constitute a basic particle in the PF algorithm. Each basic particle characterizes a possible river flood state. The probability distribution of the river flood state can be represented by the particle set. During the assimilation process, the water level observation data of each station is used to evaluate the accuracy of the prior predicted water level value at the calculation section. The fractional likelihood weight of each station is then computed as:

$$w_{i,j,t} = \frac{1}{2\pi\sigma_t} \exp\left(-\frac{(Z_{i,j,t} - Y_{j,t})^2}{2\sigma_t^2}\right) \quad (2)$$

where  $w_{i,j,t}$  and  $Z_{i,j,t}$  represent the fractal weight and the prior predicted water level value of the  $i$ th particle at the  $j$ th station at time  $t$ ,  $Y_{j,t}$  represents the observed water level at the  $j$ th station at time  $t$ , and  $\sigma_t$  represents the standard deviation of the likelihood function at the  $j$ th station at time  $t$ , which can be reduced to constants for different stations and at different times [22].

However, the main deficit of the PF is its degradation. It is possible that most particles carry less weight and only few particles are effective after updating the particle weights, which is called “particle degeneracy” [23]. Typically, whether particles in the filter are degraded is assessed through effective number of particles [24]:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_{i,t})^2} \quad (3)$$

where  $N_{\text{eff}}$  represents the effective number of particles. When  $N_{\text{eff}}$  is below a specified value, resampling is required. Resampling as proposed by Gordon et al. [9] can solve this problem. The aim of such resampling is the elimination of less-weighted particles and copy more heavily-weighted particles. The resampling algorithm samples the set of particles  $\{x_{i,t}, w_{i,t}\}_{i=1}^N$  by the probability  $w_{i,t}$  for  $N$  times and obtains a new set of particles  $\{x_{j,t}, 1/N\}_{j=1}^N$  after sampling. Finally, the posterior probability distribution of the state is obtained by a weighting calculation.

Random noise is added to the resampled particle set to properly perturb the data to ensure particle diversity. The proposed method mainly disturbed the roughness parameter:

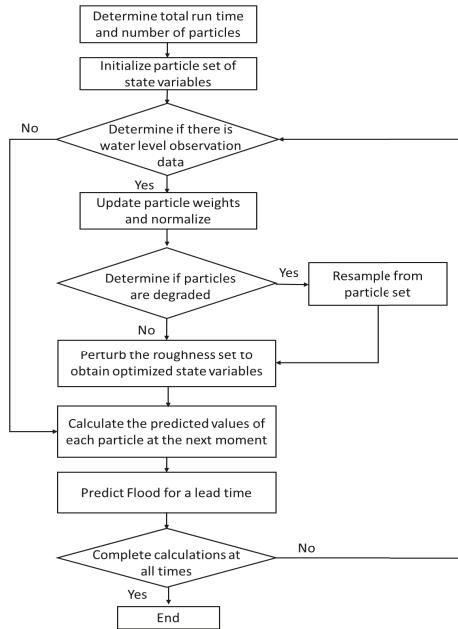
$$\hat{n}_{i,t} = n_{i,t-1} + \zeta, \quad \zeta \sim N(0, 0.004^2) \quad (4)$$

### 2.3.3. Coupling of Hydrodynamic Model and Assimilation Model

The hydrodynamic assimilation system is built by integrating the one-dimensional hydrodynamic model and the PF algorithm. Therefore, the real-time flood routing model of the river channel is mainly divided into two phases. The first phase is the hydrodynamic model forecasting phase. The flood routing process in each particle state is calculated according to known future time boundary conditions. The second phase is the PF correction phase, which updates the particle set at the moment when the observation data becomes available. Furthermore, prior distributions of particles are corrected to the posterior distributions whose mean value are closer to the real process. Both phases are interdigitated and coupled. The specific coupling procedure is shown in Figure 2 and is described in the following:

- (1) At the initial time, a random sample from the initial distribution of the water level, flow discharge, and roughness of each cross-section along the river as well as a set of  $N_p$  particles with equal weights are obtained.
- (2) The hydrodynamic model is used to calculate the water level and flow discharge at the present time step for each particle and a set of calculation results is obtained for all particles.
- (3) The assimilation module is used for present time step if the water level observation data is available. The weight of the particle set is updated using Equation (2) and referring to the observed water level data. The particles with prior water level values closer to the measured water level values obtain a larger weight calculated by a likelihood function. If the updated particles are degraded, resampling is initiated using the method proposed by Gordon et al. [9] and the roughness coefficient is perturbed using Equation (4).
- (4) The assimilation module is exited and the procedure starts at Step (2), continuing the hydrodynamic model calculation for the next time step.

Finally, the water level, flow discharge, and roughness calculated at the present time step are completely different from those of the hydrodynamic model without PF. The state of the water flow and the drag coefficient are corrected, which in turn affects all subsequent calculations. Through continuous model coupling and correction, the prediction results are closer to the true value.



**Figure 2.** Flow chart of the hydrodynamic assimilation system.

## 2.4. Model Setup and Evaluation Index

### 2.4.1. Numerical Model Setup

The topographical data used for the numerical model were consistent with the micro-model and had a total of 57 computational cross-sections. The boundary condition was set to the measured water discharge at the inlet and a synchronous water level at outlet. The initial conditions were set to the discharge and water level of each section determined by the boundary condition at the first moment. The time step was set to 1 s to satisfy Courant stability conditions. The initial roughness coefficient of the river was calibrated using the experiments of the calibration group and was set to the calibrated result of 0.05.

For the assimilation model, the initial discharge, water level, and roughness of each particle were derived by perturbing the initial values of each cross-section along the river. Furthermore, the perturbing noises were  $0.01 Q_0$ , 0.001 mm and 0.002, respectively. The perturbation of the boundary conditions was similar.

### 2.4.2. Model Performance Evaluation Index

The root mean square error (*RMSE*) and the mean relative error (*MARE*) were used to quantitatively describe the accuracy of the particle set mean. The specific definition is shown in the following:

$$RMSE = \sqrt{\frac{1}{M} \sum_{m=1}^M (\bar{x}_m - z_m)^2} \quad (5)$$

$$MARE = \frac{1}{M} \sum_{m=1}^M \frac{|\bar{x}_m - z_m|}{z_m} \quad (6)$$

where  $M$  represents the number of observations,  $z_m$  represents the  $m$ th observation value, and  $\bar{x}_m$  represents the mean value of the particle set corresponding to the  $m$ th observation time. Smaller calculation results of  $RMSE$  and  $MARE$  indicate higher accuracy of the simulation results.

The quantile-quantile plot (Q-Q plot) was also used as a visual method with which to examine the reliability of the probability distribution. In the Q-Q plot, the  $x$ -axis is a theoretic quantile of a uniform distribution and the  $y$ -axis is the actual quantile of the observation. If the plot follows a 1:1 line, the probabilistic prediction is perfectly reliable. A detailed description of the Q-Q plot can be found in Laio and Tamea [25]. The Q-Q plot has a corresponding quantitative indicator-reliability  $\alpha$ :

$$\alpha = 1 - 2 \sum_{m=1}^M \frac{|y_m - \frac{m}{M}|}{M} \quad (7)$$

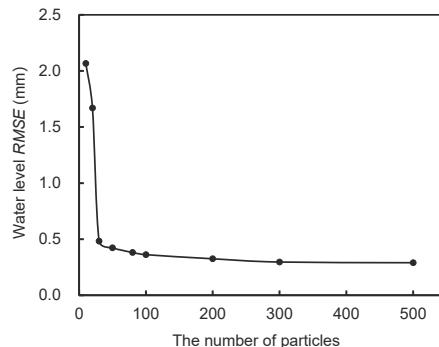
where  $y_m$  represents the normalized observation quantile. The reliability  $\alpha$  ranges between 0 and 1. A larger  $\alpha$  value indicates a better degree of coincidence of the quantile-quantile curve with the 1:1 line, and better reliability of the probability distribution interval of the particle set.

### 3. Results and Discussion

#### 3.1. Model Correction Performance

##### 3.1.1. Determination of the Number of Particles

In the assimilation model, a larger number of particles indicates better correction performance. However, with increasing number of particles, the computation load also increases greatly. Sensitivity analysis on particle number was conducted by choosing one of the experiments. Figure 3 shows the relationship between the number of particles and the water level root mean square errors. The water level  $RMSE$ s decreased with increasing number of particles. Especially, the  $RMSE$ s decreased sharply when the number of particles was below 50. To ensure the simulation accuracy of the model and short calculation time, the number of particles was set to 100.

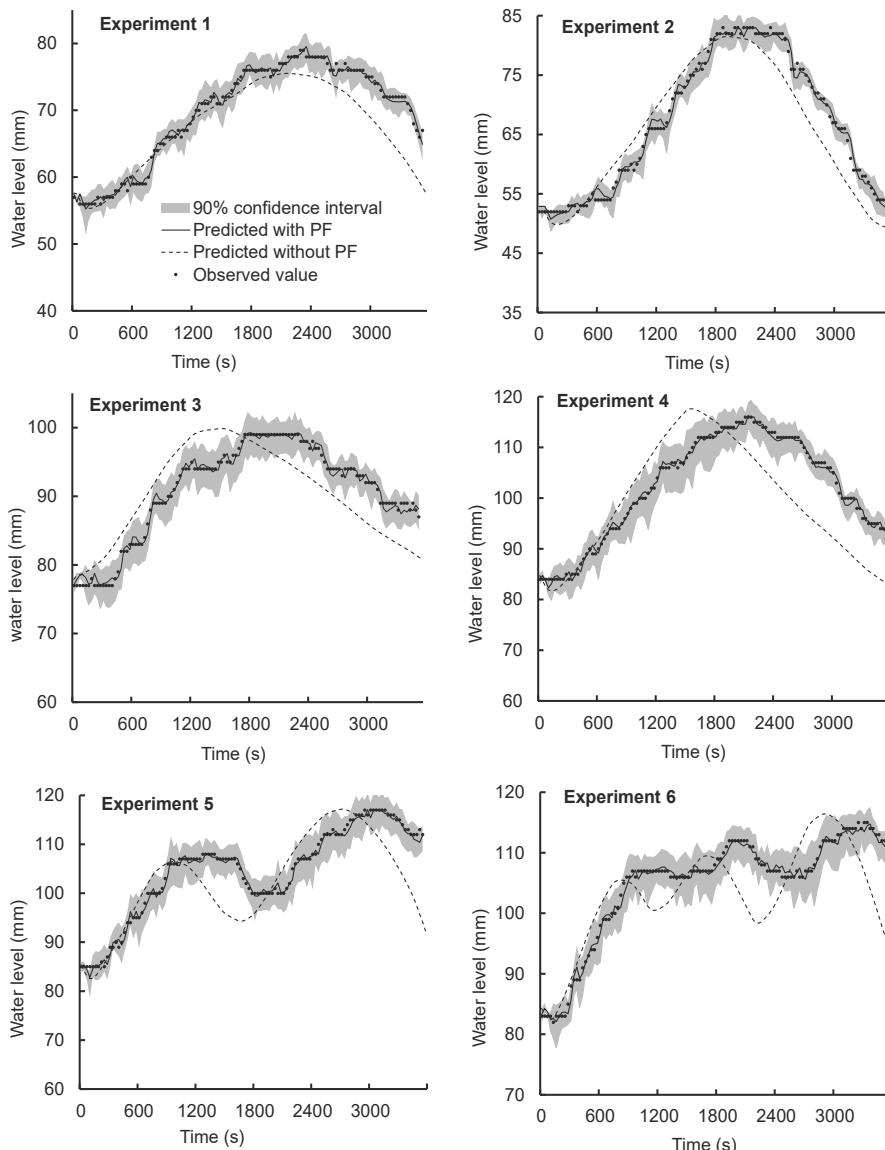


**Figure 3.** Relationship between the number of particles and the water level  $RMSE$ .

##### 3.1.2. Correction of Water Level and Roughness Coefficient

The observed water levels of the last six verification experiments were chosen to verify the correction performance of the assimilation model. The total run time of each experiment was 1 h and the assimilation correction time step was consistent with the water level acquisition time step (both 10 s). Figure 4 shows a comparison between observed and predicted water level processes both with PF and without PF (only traditional hydrodynamic model) for the experiments. Figure 4 shows that the predicted water level without PF deviates greatly from the observed curve with increasing

time, while the level predicted with PF is closer to the observed value. This is because the assimilation model incorporates the observed value in real-time, and parameters of the model are constantly optimized to let the result approach the actual value. In addition, many values fall within the 90% confidence intervals.



**Figure 4.** Comparison between observed and predicted water level processes. The dots represent the observed values. The black line represents the mean of the particles simulated by assimilation model. The dotted line represents the water level calculated by the hydrodynamic model. The gray shaded area represents the 90% confidence interval.

Table 1 shows the *RMSEs* and *MAREs* of water levels predicted with and without PF for different experiments. The *RMSE* of each experiment after assimilation was close and small. Compared to the results without PF, the *RMSEs* of the predicted value with PF decreased by 75%, 80%, 81%, 90%, 87% and 86%, respectively. The *MAREs* of the predicted value with PF also decreased by 73%, 82%, 89%, 81%, 86% and 86%, respectively. The reliability of the predicted results was also calculated and found to be above 0.8. More than 90% of the observed values fall into the 90% confidence interval, which illustrates that the assimilated results are reliable. The results shown in Figure 4 and Table 1 indicate that the assimilation model can effectively improve the obtained prediction accuracy.

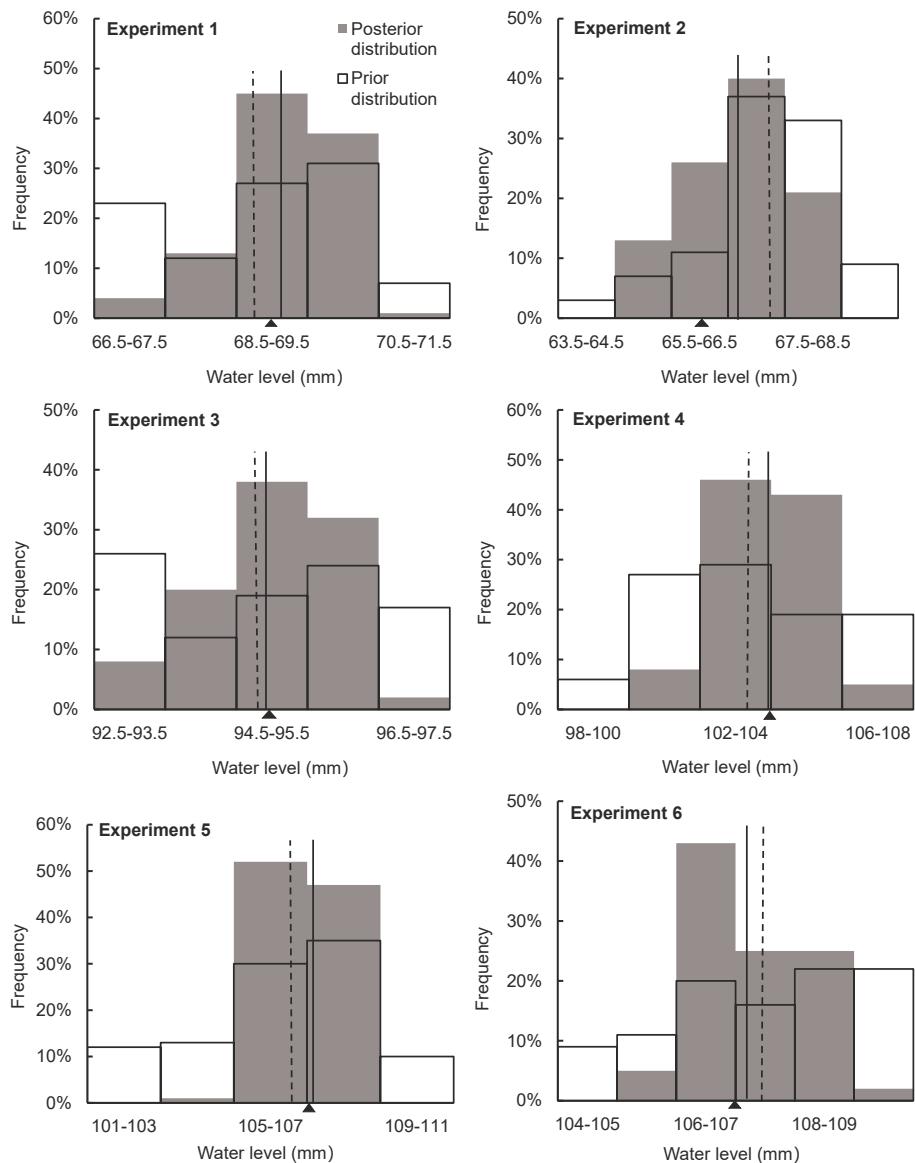
**Table 1.** Correction performance for different experiments.

Experiments	Water Level <i>RMSE</i> (mm)		Water Level <i>MARE</i> (mm)		Reliability $\alpha$	90% Confidence Interval
	Without PF	PF	Without PF	PF		
Experiment 1	2.65	0.66	2.80%	0.75%	0.86	97.5%
Experiment 2	3.67	0.75	4.77%	0.84%	0.93	91.6%
Experiment 3	4.27	0.79	4.41%	0.50%	0.86	98.3%
Experiment 4	6.90	0.70	5.90%	1.10%	0.83	99.2%
Experiment 5	5.83	0.78	4.17%	0.59%	0.88	98.3%
Experiment 6	5.62	0.77	4.06%	0.58%	0.85	97.5%

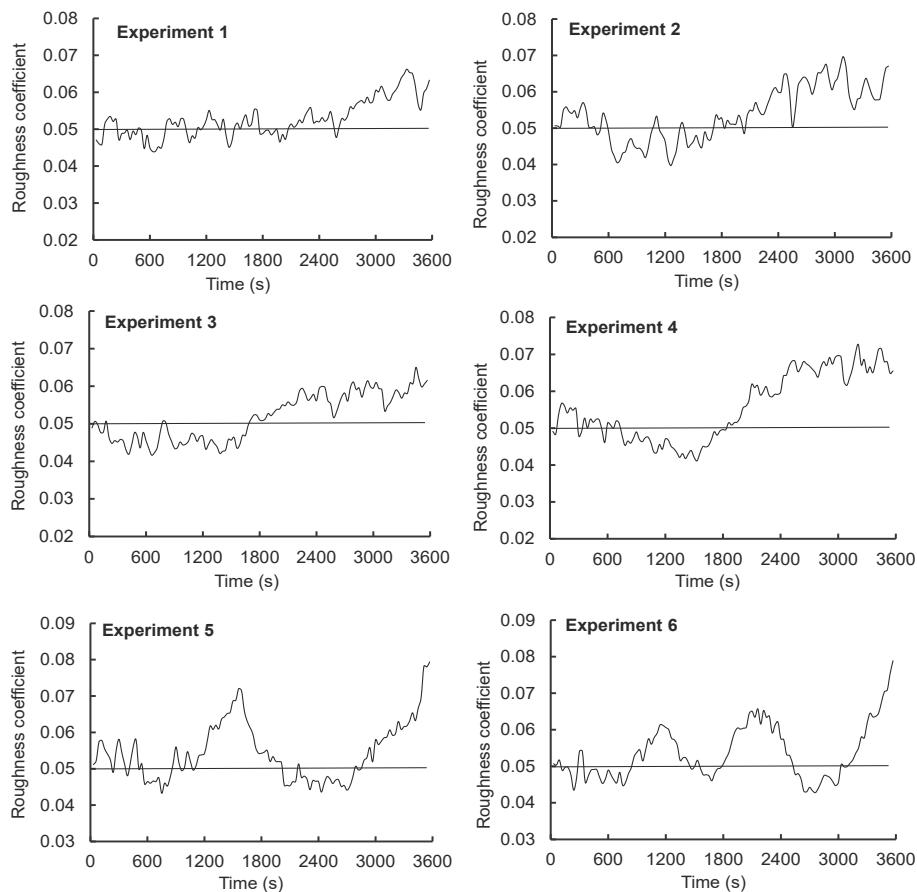
The probability distribution of the water level simulated at each time step was also analyzed. Since the water level changes greatly at  $t = 40$  min, the prior (before assimilation) and posterior (after assimilation) distributions of the water level were compared at this time step to access the performance of the assimilation model at this critical time. The correction result of the water level at this particular time step can better represent the performance of the model. Figure 5 shows histograms of the prior and posterior distributions of the water level of these six verification experiments.

Figure 5 illustrates the differences between the mean value of the water level for both the prior and posterior distribution of particles. The prior distribution of the particle set is more scattered, without obvious aggregation near the observed value, while the posterior distribution of the particle set is concentrated near the observed data. Compared to the prior distribution of particles, the posterior distribution of particles is closer to the observed values. This result indicates that the PF method can effectively utilize the observation data and maintain more particles near the observation data. This also proves that this method achieves good correction performance for the water level.

Figure 6 shows the roughness retrieved from the observed data over time. The curves indicate the real-time correction processes of roughness, and the straight line indicates the fixed value used in the hydrodynamic mathematical model. Figure 6 shows that the changing process of the corrected roughness is rather complicated. Different experiments show that the roughness continuously fluctuated but changed gently in the increasing phase of the flow process. However, during the decreasing phase of the flow process, the roughness changed sharply and increased continuously. This phenomenon is more apparent in Experiments 4–6. This characteristic is consistent with previous studies on the analysis of the change in the resistance characteristics of the channel in the LYR [26]. Within a certain flow range, the roughness coefficient is inversely proportional to the flow rate. This is mainly due to the form resistance of both the side walls and the riverbed.



**Figure 5.** Histograms of prior (gray) and posterior (colorless) distributions of water levels in different experiments. The observed value is marked with a black triangle. The black dotted line indicates the mean value of the prior distribution of particles and the black solid line indicates the mean value of the posterior distribution of particles.



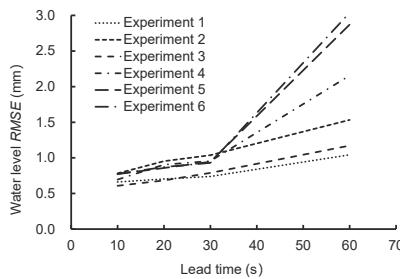
**Figure 6.** Dynamic variation process of roughness. The straight line indicates the fixed value used in the hydrodynamic model.

### 3.2. Model Prediction for Different Lead Times

As state above, the correction performance of the PF-based assimilation algorithm is good. Therefore, the corrected flow discharge, water level, and roughness were used as initial conditions for the flood prediction at the next time step. The boundary conditions of the future moments should have been predicted via the hydrological model; however, since this study focused on flood routing forecasting and its uncertainty, the historical observation data at the boundary section was appropriately disturbed as the boundary conditions. The lead times were set to 10 s, 20 s, 30 s, and 60 s.

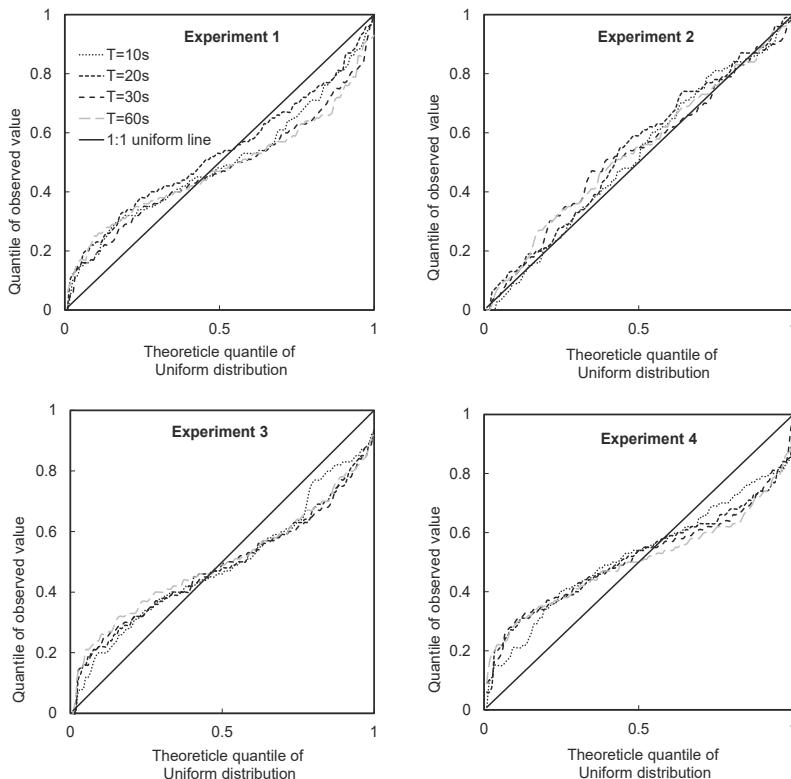
Figure 7 shows the relationship between the lead times and the RMSEs of water level predictions for different experiments. The following results were obtained: (1) Under different inflow conditions, the RMSEs of the predicted water level increases significantly with increasing lead time. (2) Comparison between the experiments of both verification groups showed that there is little difference of the RMSEs when the lead time remained below 30 s; however, the RMSEs of the second group was clearly larger than that of the first group for a lead time above 30 s. The main reason is that the roughness value used to predict the flooding process in the future is the current roughness value, which changed constantly during the actual flood process. The water flow fluctuations are more complicated in the second group and the magnitude of the change of the incoming water flow exceeds

that of the first group. The uncertainty of the roughness coefficient obviously increased with increasing lead time.

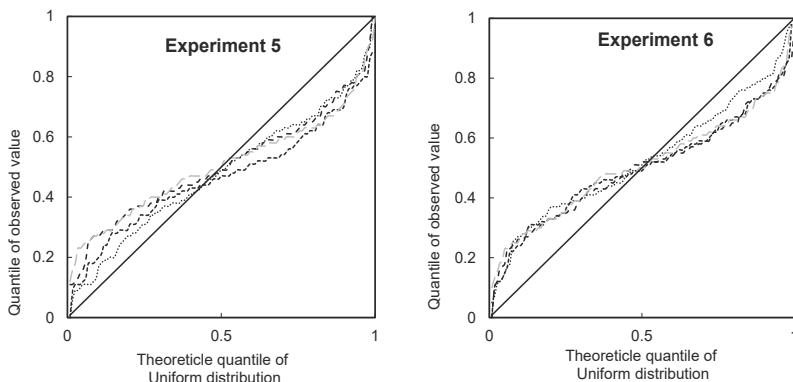


**Figure 7.** RMSEs of water level prediction for different lead times in different experiments.

Figure 8 shows the Q-Q plots of water level predictions for different experiments in response to different lead times. The Q-Q plots are close to the 1:1 uniform line for a lead time of 10 s. The fitting effect of the quantile of observed values and the 1:1 uniform line is deteriorating with increasing lead times. In general, with increasing lead time, the reliability of the probability prediction interval of the water level gradually decreases; however, it can still represent the uncertainty of the water level forecast result well.



**Figure 8. Cont.**



**Figure 8.** Q-Q plots of the water level prediction for different lead times in different experiments. T indicates the different lead times.

### 3.3. Influence of the Flooding Lead Time

Theoretically, the hydrodynamic model itself does not have a foreseeable period, and the flood propagation process can only be calculated under given boundary conditions. For the hydrodynamic model to predict the future flood propagation process, the boundary conditions need to be predicted in advance. These boundary conditions should be predicted using both the weather forecasting model and the hydrological model. However, even if the boundary conditions are correctly chosen, the predicted results of the hydrodynamic model are not necessarily satisfactory, as can be seen in the simulated results of the hydrodynamic model without PF in Figure 4. This is because the given roughness (or the corrected roughness at the previous time step) and the initial field of water flow (the result of the previous time step) cannot adequately represent the hydraulic conditions during the following period of time. When the lead time is small, the model parameters and the initial field of water flow exert less impact on the future water level forecast. However, when the lead time is sufficiently large, the given roughness and flow conditions are quite different from the actual conditions; consequently, the simulated water level results will greatly deviate.

At the same time, even if the lead time remains identical, the PF-based assimilation model achieves different degrees of improvement for the prediction results of both designed sets of experiments, compared to the hydrodynamic model (as shown in Figure 4). The correction performance of the PF algorithm is not affected by complex changes of the water conditions, which shows that the assimilation model has good applicability for water level prediction under different scenarios. It can be inferred that, under the same accuracy requirements, the PF-based assimilation model can increase the flooding lead time, thus gaining time for both flood prevention and emergency evacuation. With reference to the designed scale of the geometry, slope, and roughness of the LYR micro-model, the lead time of 30 s is equivalent to about 7.4 h in the prototype of the Yellow River. Therefore, the improvement of the flood forecasting accuracy and the increase of flooding lead time achieved by the assimilation model are of great significance for flood control and disaster relief.

### 3.4. Application to Real Rivers

It should be emphasized that the flood routing model presented here is universal and not regionally limited. It can not only be used for indoor physical models but also for various practical river channels. However, the premise is that a real-time data automatic acquisition system needs to be established on the actual river channel, and the obtained data have to be continuously transmitted to the hydrodynamic assimilation system for the prediction. The upstream and downstream boundary conditions of the future of the river also need to be specified correctly. The hydrodynamic model

developed here only simulates the flow routing without considering other forcing data. To truly improve the accuracy of flood prediction for longer lead times in an actual river channel, a complete flood forecasting system has to be established. In addition to the model presented in this paper, two other basic elements are required: (i) a rainfall forecasting model; and (ii) a rainfall–runoff forecasting model [2]. Further coupling of these models requires future research. In addition, the hydrodynamic model developed in this study needs to be further improved against actual river conditions, e.g., considering over-bank flow, sediment transport, curved circulation, and dam scheduling.

#### 4. Conclusions

This study verified and evaluated the accuracy of an assimilation model system for real-time flood prediction at micro-model scale in the laboratory. A PF-based assimilation model was coupled with a one-dimensional hydrodynamic model to simulate the water level under different boundary conditions. Real-time water level observations at observation points along the channel were assimilated into the modeling system to achieve probabilistic forecasting.

We used the assimilation model system to predict the flood process against six different experiments. All results indicate that the RMSEs and MAREs of the water level predicted by the assimilation model were much lower than the results of the hydrodynamic model without PF. These results prove the precision and accuracy of water level forecasting with the assimilation model. With increasing lead time, the RMSEs of the predicted water level increased significantly. However, the RMSEs of the results predicted by the assimilation model were much smaller than those predicted by the hydrodynamic model without PF, indicating that the assimilation model can enlarge the flooding lead time. Of course, further coupling of hydrological and weather forecasting models will increase the applicability of the assimilation model. The application of this PF-based assimilation model to flood forecasting is of great significance toward improving the accuracy of real flood forecasting in the LYR. It enables more accurate decisions for future floods without delay, thus reducing property losses on both sides of the Yellow River.

**Author Contributions:** Conceptualization, M.C.; Methodology, J.P. and M.C.; Software, J.P.; Tests: J.P. and P.W.; Writing—Original Draft Preparation, M.C.; Writing—Review and Editing, M.C.; Supervision, M.C.; Project Administration, M.C.; and Funding Acquisition, M.C.

**Funding:** This research was made possible with the support of the National Key Research and Development Program (Grant Number: 2016YFC0400107 and 2016YFC0402506).

**Acknowledgments:** We are very grateful to Hongwei Fang of Tsinghua University for providing experiment sites, water and electricity facilities. We thank the editors and the anonymous reviewers for their constructive comments, which helped us improve the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

- Wu, S.; Lien, H.; Chang, C. Modeling risk analysis for forecasting peak discharge during flooding prevention and warning operation. *Stoch. Environ. Res. Risk Assess.* **2010**, *24*, 1175–1191. [[CrossRef](#)]
- Hsu, M.H.; Fu, J.C.; Liu, W.C. Flood routing with real-time stage correction method for flash flood forecasting in the Tanshui River, Taiwan. *J. Hydrol.* **2003**, *283*, 267–280. [[CrossRef](#)]
- David, C.; Smith, G. The United States weather service river forecast system. In *Real-Time Forecasting/Control of Water Resource Systems*; Pergamon Press: Oxford, UK, 1980; pp. 305–325.
- Chaudhry, M.H. *Open-Channel Flow*; Springer: Boston, MA, USA, 2008. [[CrossRef](#)]
- Patro, S.; Chatterjee, C.; Singh, R.; Raghuvanshi, N.S. Hydrodynamic modelling of a large flood-prone river system in India with limited data. *Hydrol. Process.* **2009**, *23*, 2774–2791. [[CrossRef](#)]
- Han, D.; Fang, H.W.; Bai, J.; He, G.J. A coupled 1-D and 2-D channel network mathematical model used for flow calculations in the middle reaches of the yangtze river. *J. Hydodyn.* **2011**, *23*, 521–526. [[CrossRef](#)]

7. Chen, M.; Fang, H.; Zheng, Y.; He, G. Integrated Flood Management for Beiyun River, China. *J. Hydrol. Hydromech.* **2013**, *61*, 177–187. [[CrossRef](#)]
8. Arulampalam, M.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
9. Gordon, N.; Salmond, D.; Smith, A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F Radar Signal Process.* **1993**, *140*, 107–113. [[CrossRef](#)]
10. Van Der Merwe, R.; Doucet, A.; De Freitas, N.; Wan, E. The Unscented Particle Filter. In *Advances in Neural Information Processing Systems 13*; MIT Press: Cambridge, MA, USA, 2001; pp. 584–590.
11. Han, X.; Li, X. An evaluation of the nonlinear/non-Gaussian filters for the sequential data assimilation. *Remote Sens. Environ.* **2008**, *112*, 1434–1449. [[CrossRef](#)]
12. Pan, M. Assimilation of Satellite Observations into a Land Surface Hydrologic Modeling System. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 2006.
13. Xiong, X.; Navon, I.M.; Uzunoglu, B. A Note on the Particle Filter with Posterior Gaussian Resampling. *Tellus A Dyn. Meteorol. Oceanogr.* **2006**, *58*, 456–460. [[CrossRef](#)]
14. Kivman, G.A. Sequential parameter estimation for stochastic systems. *Nonlinear Process. Geophys.* **2003**, *10*, 253–259. [[CrossRef](#)]
15. Pham, D.T. Stochastic Methods for Sequential Data Assimilation in Strongly Nonlinear Systems. *Mon. Weather Rev.* **2001**, *129*, 1194–1207. [[CrossRef](#)]
16. Xu, X.; Zhang, X.; Fang, H.; Lai, R.; Zhang, Y.; Huang, L.; Liu, X. A real-time probabilistic channel flood-forecasting model based on the Bayesian particle filter approach. *Environ. Model. Softw.* **2017**, *88*, 151–167. [[CrossRef](#)]
17. Moradkhani, H.; Hsu, K.L.; Gupta, H.; Sorooshian, S. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resour. Res.* **2005**, *41*, 1–17. [[CrossRef](#)]
18. Xia, J.; Zhang, X.; Wang, Z.; Li, J.; Zhou, M. Modelling of hyperconcentrated flood and channel evolution in a braided reach using a dynamically coupled one-dimensional approach. *J. Hydrol.* **2018**, *561*, 622–635. [[CrossRef](#)]
19. Karaer, F.; Koparal, A.; Tombul, M. Environmental risk determination of flood in Porsuk River basin via one-dimensional modelling. *Appl. Ecol. Environ. Res.* **2018**, *16*, 4969–4983. [[CrossRef](#)]
20. van Leeuwen, P.J. Particle Filtering in Geophysical Systems. *Mon. Weather Rev.* **2009**, *137*, 4089–4114. [[CrossRef](#)]
21. Wikle, C.K.; Berliner, L.M. A Bayesian tutorial for data assimilation. *Phys. D Nonlinear Phenom.* **2007**, *230*, 1–16. [[CrossRef](#)]
22. Matgen, P.; Montanari, M.; Hostache, R.; Pfister, L.; Hoffmann, L.; Plaza, D.; Pauwels, V.R.N.; De Lannoy, G.J.M.; De Keyser, R.; Savenije, H.H.G. Towards the sequential assimilation of SAR-derived water stages into hydraulic models using the Particle Filter: Proof of concept. *Hydrol. Earth Syst. Sci.* **2010**, *14*, 1773–1785. [[CrossRef](#)]
23. Snyder, C.; Bengtsson, T.; Bickel, P.; Anderson, J. Obstacles to High-Dimensional Particle Filtering. *Mon. Weather Rev.* **2008**, *136*, 4629–4640. [[CrossRef](#)]
24. Cao, B.; Ma, C.; Liu, Z. A Fine Resampling algorithm for general particle filters. In Proceedings of the 2011 4th International Congress on Image and Signal Processing, Shanghai, China, 15–17 October 2011; Volume 5, pp. 2695–2699. [[CrossRef](#)]
25. Laio, F.; Tamea, S. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrol. Earth Syst. Sci.* **2007**, *11*, 1267–1277. [[CrossRef](#)]
26. Shang, H.; Han, M.; Liu, Q.; Zhou, W.; Pei, M. Analysis of Hydraulic Dynamic Conditions and Resistance Characteristics Change in the Lower Yellow River Beach River beach area. *Yellow River* **2000**, *22*, 9–10. [[CrossRef](#)] (In Chinese)



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Application of Artificial Neural Networks for Accuracy Enhancements of Real-Time Flood Forecasting in the Imjin Basin

Aida Jabbari and Deg-Hyo Bae \*

Department of Civil & Environmental Engineering, Sejong University, 98 Gunja-Dong, Gwangjin-Gu, Seoul 143-747, Korea; ajabbari@sju.ac.kr

\* Correspondence: dhbae@sejong.ac.kr

Received: 31 August 2018; Accepted: 9 November 2018; Published: 11 November 2018

**Abstract:** Hydrometeorological forecasts provide future flooding estimates to reduce damages. Despite the advances and progresses in Numerical Weather Prediction (NWP) models, they are still subject to many uncertainties, which cause significant errors forecasting precipitation. Statistical postprocessing techniques can improve forecast skills by reducing the systematic biases in NWP models. Artificial Neural Networks (ANNs) can model complex relationships between input and output data. The application of ANN in water-related research is widely studied; however, there is a lack of studies quantifying the improvement of coupled hydrometeorological model accuracy that use ANN for bias correction of real-time rainfall forecasts. The aim of this study is to evaluate the real-time bias correction of precipitation data, and from a hydrometeorological point of view, an assessment of hydrological model improvements in real-time flood forecasting for the Imjin River (South and North Korea) is performed. The comparison of the forecasted rainfall before and after the bias correction indicated a significant improvement in the statistical error measurement and a decrease in the underestimation of WRF model. The error was reduced remarkably over the Imjin catchment for the accumulated Mean Areal Precipitation (MAP). The performance of the real-time flood forecast improved using the ANN bias correction method.

**Keywords:** ANN; hydrometeorology; flood forecasting; real-time; postprocessing

## 1. Introduction

Climate change has increased extreme rainfall events, and as a result, damage from floods has significantly increased. Heavy rainfalls occurring over different areas often lead to various flooding problems. Therefore, societies need to improve flood risk management. Hydrometeorological forecasts provide future estimates to reduce damage and provide warnings of extreme events. Coupling numerical weather prediction (NWP) and hydrological models allows meteorology and hydrology connection to generate real-time flood forecasting. Real-time flood forecasting has been investigated worldwide in previous studies using hydrometeorological data [1–3].

Although NWP models have improved significantly, the restrictions of the physical processes in NWP models lead to unavoidable errors in the forecasting of meteorological properties due to nonlinearity and the chaotic atmospheric system. In analyses of the accuracy of the forecasted values of atmospheric models, the most frequent errors are related to the model error, initialization, boundary conditions of atmosphere, land and sea surface, model formulation and model parametrization during the forecast period [4]. These errors influence the output of the meteorological models.

NWP models are restricted to representing the physical processes of the chaotic atmosphere. A precise forecast of precipitation is one of the challenges of NWP models. The biased output of these models is still a concern for hydrometeorological prediction studies [5]. NWP models

are subject to many uncertainties, which cause significant errors in the forecasting of real-time precipitation. In coupled hydrometeorological studies, one critical issue before running a rainfall-runoff model is to reduce the input forcing errors that are produced by the meteorological model. Therefore, postprocessing of the model outputs of the real-time forecast data would result in a better match with the observation records. Statistical postprocessing methods improve real-time forecast accuracy by relating the model outputs to the observed values.

Several studies have shown that statistical postprocessing improved forecast performance by reducing systematic biases [6,7]. The main purpose of statistical bias correction is to develop a relationship between the modeled and observed data. Commonly used statistical methods are the quantile-based mapping method [8] and regression approaches that include linear relationships [9] and nonlinear relationships [10]. In real-time flood forecasting, a comparison of different postprocessing methods such as Bayesian Model Averaging (BMA), classic poor man ensemble (PME) and Multimodel SuperEnsemble Dressing (MSD) indicated that the MSD approach provided better precipitation data for floods in Italy [11]. Six different bias correction methods (including linear scaling (LS), local intensity scaling (LOCI) scaling, Daily Translation (DT), daily bias correction (DBC), quantile mapping based on an empirical distribution (QME) and quantile mapping based on a gamma distribution (QMG)) were applied in ten North American river basins to determine the sensitivity of the bias correction methods on climate models [12]. All bias correction methods improve the precipitation forecasts, but DT, DBC, QME, and QMG resulted in the most significant improvements to the simulations [12].

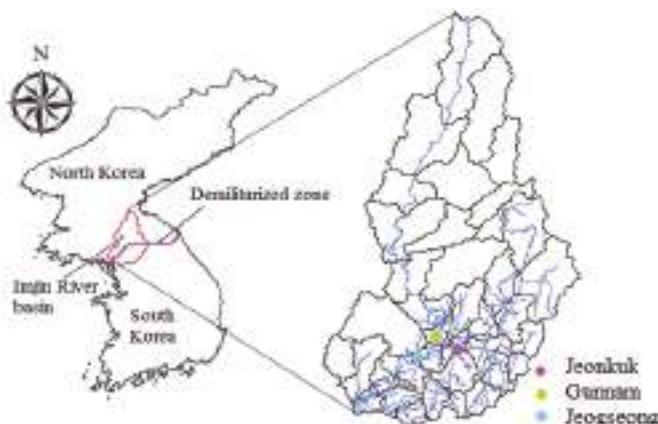
An Artificial Neural Network (ANN) can predict and modeling input and output data [5]. ANN is more practical than other techniques due its ability to handle complex nonlinear systems. In previous studies, ANN was applied for various purposes such as precipitation estimation [13–15], hydrological modeling [16–19], hydrometeorological studies [20,21], flood forecasting [22–24] and flood inundation [25,26]. Three new hybrid artificial intelligence optimization models (adaptive neuro-fuzzy inference system (ANFIS) with cultural (ANFIS-CA), bees (ANFIS-BA), and invasive weed optimization (ANFIS-IWO) algorithms) were presented for flood susceptibility mapping in Iran. The results showed that ANFIS-IWO had better performance and ANFIS-BA exhibited the better prediction capability [27]. The application of ANN for the prediction of water resource variables has been reviewed in 43 papers [28]; in addition, 210 papers (from 1999 to 2007) in which ANN was developed for river flow prediction including quantity and quality were reviewed [29]. ANN was applied to predict seasonal rainfall for the next 5 years and a multilayer neural network to predict rainfall time series was employed successfully in India [30]. The bias correction of the climate modeled temperature and precipitation was carried out using ANN over northern South America; compared with linear regression, ANN had a superior performance [5]. Recent reviews on the application of ANN have been reported in [31].

There is a lack of studies in the literature on the quantification of the accuracy improvement of coupled hydrometeorological models using ANN for the bias correction of real-time rainfall forecasts. The aim of this study is to evaluate the real-time bias correction of precipitation data from a hydrometeorological point of view. In addition, an assessment of forecast improvements to hydrological models that result in real-time flood forecasting of coupled hydrometeorological models is performed. To accomplish this aim, a variety of tests are conducted to quantify the accuracy assessment of real-time precipitation and flood forecasts using coupled hydrometeorological models. The present study provides details regarding the real-time precipitation accuracy improvement in addition to the forcing of the bias-corrected rainfall as input to the hydrological model. An evaluation of the accuracy enhancement of real-time flood forecasting of hydrometeorological models is also given.

## 2. Study Area and Data

### 2.1. Imjin Basin

The target area for this study is the Imjin River basin, which passes through the demilitarized zone (DMZ) in North and South Korea between  $37^{\circ}44'23''$  N  $126^{\circ}31'19''$  E and  $39^{\circ}11'12''$  N  $127^{\circ}36'21''$  E. The length and the area of the Imjin River watershed are 273.5 km and  $8139 \text{ km}^2$ , respectively. In addition, 62.9% of the area of the basin is in North Korea, and the other 37.1% is in South Korea. Because approximately two-thirds of the Imjin River is in North Korea, it is considered a transboundary river. Details regarding the study catchment are mapped in Figure 1. The source of the Imjin River is in North Korea, it then passes through South Korea where it joins the Han River, and it finally empties into the Yellow Sea. The Imjin River has complex topography with an altitude range from 155 m to 1570 m with a mean elevation of 680.5 m above mean sea level. The average annual rainfall is approximately 1100 mm, which mostly occurs during the late summer and fall [32]. The study area consists of 38 sub-basins that are defined by characteristics such as elevation, direction of the streamflow and soil conditions of the entire basin.



**Figure 1.** Location, sub-basins, river network and water level gauging stations of the Imjin River basin.

### 2.2. Hydrological and Meteorological Data

The Imjin River has experienced various flash floods during past years. The most extreme events are selected for consideration for flood forecasting in the Imjin River basin. In 2002, Typhoon Rusa ripped through South Korea in the Gangneung area between 31 August and 1 September, affecting the eastern and southern parts of Korea with almost 1000 mm of rainfall in 30 h [33,34]. The typhoon caused the submergence of 9000 houses and killed 113 people. In 2002 event, continuous low and high humid east wind led to the creation of frontal precipitation by cold air [34]. In 2007, Typhoon Wipha in North Korea brought heavy rain between 7–14 August and 18–20 September. Approximately 500–700 mm of rainfall caused flooding in North Korea for seven days. The westward-travelling Typhoon Wipha in 2007, caused heavy frontal precipitation was accompanied by strong wind that was developed by mesoscale disturbances in China [35]. On July 27, 2011 heavy rainfall led to extreme flooding in Seoul, South Korea. In Seoul, as a mega city with a high population concentration, 536 mm of rainfall over three days resulted in severe damage, 980 flood victims, 62 people reported as dead or missing and 19,215 inundated houses [36]. The conventional rainfall in 2011 was developed by convective systems which moved from the Yellow Sea to Seoul [37]. The percentage of the liquid precipitation in the study period was 100%.

The estimated flood damage costs due to intense rainfall events indicate the need for integrated flood management for disaster prevention. This is especially important for two countries with

different natural environments, national defenses, and political problems. The Imjin River is the seventh longest river in the Korean peninsula and encountered severe flooding during past years. Therefore, the hydrological and meteorological models are coupled to provide information to prepare for severe events, provide early alerts for imminent disasters and to minimize the flood induced hazards in this important area. Other applications of the flood forecasting system are primarily related to military operations. The Imjin River basin has a military region named Paju-si, which includes river-crossing operations. Flood hazards directly influence military operations and require the use of detailed and widely applicable hydrological models.

In this study, hydrometeorological components are coupled for real-time rainfall-runoff forecasting for the transboundary Imjin River. The observation data used in this study underwent a quality control procedure, which verified the values and filled in any missing values by the Inverse Distance Weighting (IDW) method, to complete the hourly data from at all stations.

The IDW formula is as follows:

$$R(s_0) = \sum_{i=1}^n w_i \times R(s_i) \quad (1)$$

$$w_i = \frac{d_{i0}^{-2}}{\sum_{i=1}^n d_{i0}^{-2}} \quad (2)$$

where  $R(s_0)$  is the revised rainfall (mm) at  $s_0$ ,  $R(s_i)$  is the observed rainfall (mm) at  $s_i$ ,  $n$  is the number of observed stations,  $w_i$  is the weight of  $s_i$  and  $d_{i0}$  is distances from the points to the point estimated. IDW is a deterministic spatial interpolation and one of the most popular methods of interpolation.

### 3. Methodology

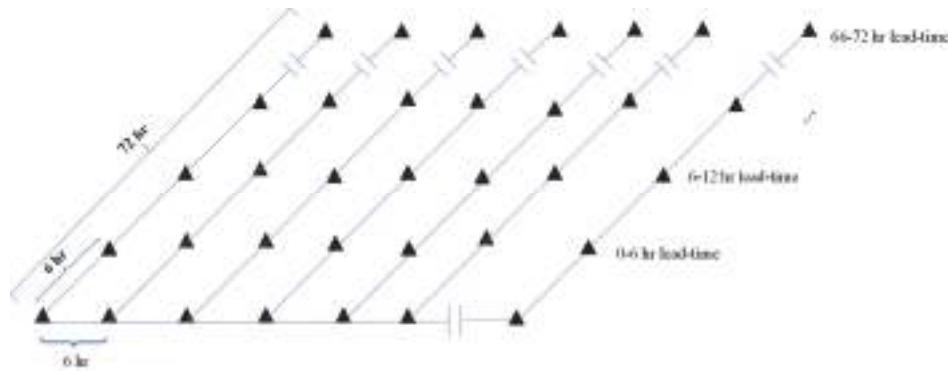
#### 3.1. WRF Model

The WRF model has been developed as a mesoscale model for research and operational NWP model studies. The WRF model can be used for different aspects of atmospheric and operational forecasting research at various scales such as forecasting extreme events and is especially useful for heavy rainfall predictions at a high spatial resolution [38,39]. The WRF model is a very useful tool for numerical weather predictions and data assimilation in estimating meteorological data such as rainfall, solar radiation, temperature, dew point, and wind speed by reproducing climate features at ultrafine temporal and spatial resolutions [40].

The initial and boundary conditions were obtained using external sources, such as the static geographic data provided by the United States Geological Survey (USGS) and Moderate Resolution Imaging Spectroradiometer (MODIS) data set and the gridded data provided by regional and global models including the North American Mesoscale Forecast system (NAM) and the Global Forecast System (GFS) [41]. The weather charts employed in this study were provided by the National Centers for Environmental Prediction (NCEP) Final Analysis (FNL) and have a resolution of  $1^\circ \times 1^\circ$ . In summary, using the definitions of all computational grids, geogrid interpolates terrestrial time-invariant fields, then Ungrid extracts the meteorological fields from the GRidded Information in Binary (GRIB) formatted files, and Metgrid horizontally interpolates the meteorological data to the simulation domains. Time-splitting techniques are used by the Advanced Research WRF (ARW) solver to integrate the fully compressible nonhydrostatic equations of motion. The Euler equations are in flux form and are formulated using a terrain that follows mass vertical coordinates. Finally, the second- or third- order Runge-Kutta method is applied for time-split integration [41].

In this study, WRF version 3.5.1 was applied to perform real-time forecasting of the meteorological data by using the WRF Double-Moment 6-Class (WDM6) microphysical scheme. The WRF model provided an ultrafine scale of temporal and spatial resolution (10 min and  $1\text{ km} \times 1\text{ km}$  respectively) that covers Korea and the surrounding region. The real-time forecast data had a 10-min temporal

resolution over a 72-h time series which was repeated every 6 h. The schematic construction of the real-time forecast of the WRF model is shown in Figure 2.



**Figure 2.** The schematic construction of the real-time forecast of the WRF model.

### 3.2. Sejong University Rainfall-Runoff (SURR) Model

The SURR model was developed by the Water Resource and GIS Laboratory, Sejong University [42] and is based on the event-oriented storage function model (SFM) by [43]. The SURR considers the rainfall loss using a soil moisture accounting model for streamflow simulation and prediction. The SURR is a semi-distributed continuous rainfall-runoff model that uses physical foundations to estimate the hydrological components including soil moisture and surface, lateral and groundwater in conjunction with the basin and channel routing to simulate the runoff. The governing equations of the SURR model for flow generation are as follows.

$$\frac{dS_{sb}(t)}{dt} = Q_{sur}(t) + Q_{lat}(t) + Q_{gw}(t) \quad (3)$$

$$Q_{sur}(t) = (Q'_{sur}(t) + Q_{surstor}(t - 1))(1 - \exp[-surlag]) \quad (4)$$

$$Q_{lat}(t) = (Q'_{lat}(t) + Q_{latstor}(t - 1))(1 - \exp[-latlag]) \quad (5)$$

$$Q_{gw}(t) = Q_{gw}(t - 1) \cdot \exp[-\alpha_{gw} \cdot \Delta t] + W_{rchrg}(t) \cdot (1 - \exp[-\alpha_{gw} \cdot \Delta t]) \quad (6)$$

$$Q_{sb}(t) = \left( \frac{S_{sb}(t)}{K_{sb}} \right)^{\frac{1}{P_{sb}}} \quad (7)$$

$$\frac{dS_{ch}(t)}{dt} = \sum \left( \frac{S_{ch}(t)}{K_{ch}} \right)^{\frac{1}{P_{ch}}} - Q_{ch}(t) \quad (8)$$

$$Q_{ch}(t) = \left( \frac{S_{ch}(t)}{K_{ch}} \right)^{\frac{1}{P_{ch}}} \quad (9)$$

where  $S_{sb}(t)$  and  $Q_{sb}(t)$  are the storage content and outflow of the storage in the basin and  $S_{ch}(t)$  and  $Q_{ch}(t)$  the storage content and outflow of the channel at time  $t$ , respectively.  $Q_{sur}(t)$  and  $Q'_{sur}(t)$  are the surface runoff with and without the lag effect consideration at time  $t$ ;  $Q_{surstor}$  is the surface runoff which stored in the watershed at time  $t - 1$  and  $surlag$  is surface runoff lag coefficient.  $Q_{lat}(t)$  and  $Q'_{lat}(t)$  are the lateral flow with and without the lag effect consideration at time  $t$ ;  $Q_{latstor}$  is the lateral flow which stored in the watershed at time  $t - 1$  and  $latlag$  is lateral flow lag coefficient.  $Q_{gw}(t)$  is groundwater flow contributing to the discharge from each sub-basin at time  $t$ ;  $\alpha_{gw}$  base flow recession

constant;  $W_{rchr}(t)$  is recharge entering the aquifer and  $\Delta t$  is the calculation time interval.  $K_{sb}$ ,  $P_{sb}$ ,  $K_{ch}$  and  $P_{ch}$  are the storage function constants in the basin and channel, respectively.

The input of the SURR model is MAP and Mean Areal Evapotranspiration (MAE) for each of the sub-basins. The observed and forecasted meteorological data can be applied to drive the SURR model for streamflow simulation and forecasting, respectively. The evapotranspiration is calculated using the FAO Penman-Monteith (PM) formula. The FAO PM method is recommended as a standard method for estimating evapotranspiration (ET). The FAO PM method can be expressed as Equation (1):

$$ET = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T+273} u_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)} \quad (10)$$

where ET is the evapotranspiration ( $\text{mm day}^{-1}$ ),  $R_n$  is the net radiation at the crop surface ( $\text{MJ m}^{-2} \text{ day}^{-1}$ ),  $G$  is the soil heat flux density, which is relatively small for daily and ten-day periods ( $\text{MJ m}^{-2} \text{ day}^{-1}$ ),  $T$  is the mean daily air temperature at a height of 2 m ( $^{\circ}\text{C}$ ),  $u_2$  is the wind speed at a height of 2 m ( $\text{m s}^{-1}$ ),  $e_s$  is the saturation vapor pressure (KPa),  $e_a$  is the actual vapor pressure (KPa),  $e_s - e_a$  is the saturation vapor pressure deficit (KPa),  $\Delta$  is the slope vapor pressure curve ( $\text{KPa } ^{\circ}\text{C}^{-1}$ ), and  $\gamma$  is the psychrometric constant ( $\text{KPa } ^{\circ}\text{C}^{-1}$ ). The meteorological data are used to calculate the ET, and then, Thiessen polygons are used by GIS to estimate the MAE for each sub-basin. The rainfall and evapotranspiration data have hourly temporal resolutions, which were spatially interpolated by the Thiessen polygons method using GIS. A detailed description of the SURR model is reported in [42].

The hydrologic models simulate the response of the basin to a given rainfall. Therefore, as an initial assessment, the calibration and verification of the hydrological model could be performed to ensure that the results are accurate and stable. The SURR model was calibrated for the Imjin basin using the observed rainfall and streamflow, and the optimized parameters resulted in a good agreement between the observed and simulated streamflow during the verification periods. The Nash-Sutcliffe Efficiency (NSE) by [44], the Relative Error in Volume (REV) and Kling-Gupta Efficiency (KGE) proposed by [45] were used to compare the results in the calibration and verification periods.

The NSE, REV and KGE equations (following the KGE equation represented in [46]) are as follows:

$$NSE = 1 - \frac{\sum_{i=1}^n (O_i - S_i)^2}{\sum_{i=1}^n (O_i - \bar{O})^2} \quad (11)$$

$$REV = \frac{\sum_{i=1}^n S_i - \sum_{i=1}^n O_i}{\sum_{i=1}^n O_i} \times 100 \quad (12)$$

$$KGE = 1 - \sqrt{(r - 1)^2 + (\alpha - 1)^2 + (\beta - 1)^2} \quad (13)$$

$$\alpha = \frac{\sigma_s}{\sigma_o} \quad (14)$$

$$\beta = \frac{\mu_s}{\mu_o} \quad (15)$$

where  $O_i$  is the observed data,  $S_i$  is simulated data,  $\bar{O}$  is the average of observed data,  $r$  is the correlation,  $\alpha$  is the ratio of the simulation to the observed streamflow standard deviation and  $\beta$  is the ratio of the mean of the simulated and observed streamflow.

The NSE is selected to evaluate the performance of the hydrological model since it shows goodness of fit for the hydrological model. The NSE is a model performance evaluation criterion which is used worldwide. However, the NSE calculates the differences between the observed and simulated streamflow in squared values. This may lead to over- and underestimation of model performance during the high and low flow respectively [47]. Therefore, the REV efficiency criterion is also selected to measure the ratio of the absolute error of the simulated and observed data to the observed data. To overcome the model skill overestimation in NSE criterion the KGE as an alternative metric is

selected to measure the hydrological model performance. A comparison between NSE and KGE is discussed in [45]. The above-mentioned criteria are used to indicate whether the results of the model simulation in calibration and verification process are reasonable or not.

With the NSE range set between 1 (i.e., the ideal value) and negative infinity, values lower than zero indicate that the mean value of the observed streamflow could have better estimate than the model provides. According to the calibration and verification of the SURR model, the results of the streamflow simulations are reasonable and stable with NSEs close to 1. The REV is a measure of precision; it is the ratio of the absolute error of the simulated and observed data to the observed data. The REV can vary among negative infinity to positive infinity and the ideal value is zero. In this study, the results of the calibration and verification for the values of the NSE, REV and KGE showed that the SURR model can well reproduce the observed streamflow in Gunnam, Jeonkuk and Jeogseong stations. The results of the calibration and verification are shown in Table 1.

**Table 1.** Statistical analysis for the simulated discharge for the calibration and verification periods in the SURR model.

Metric	Calibration Period 23 July–4 September 2007			Calibration Period 1 July–22 August 2008			Verification Period 21 June–4 August 2009		
	Gunnam	Jeonkuk	Jeogseong	Gunnam	Jeonkuk	Jeogseong	Gunnam	Jeonkuk	Jeogseong
NSE	0.69	0.78	0.71	0.70	0.83	0.79	0.57	0.85	0.79
REV	-0.48	-0.12	-0.52	0.37	0.03	0.08	0.16	-0.22	0.03
KGE	0.53	0.62	0.51	0.47	0.85	0.69	0.75	0.68	0.80
Metric	Verification Period 9 July–20 August 2010			Verification Period 16 June–2 August 2011			Verification Period 31 July–13 September 2012		
	Gunnam	Jeonkuk	Jeogseong	Gunnam	Jeonkuk	Jeogseong	Gunnam	Jeonkuk	Jeogseong
NSE	0.62	0.71	0.67	0.71	0.89	0.85	0.59	0.78	0.66
REV	0.23	-0.34	-0.07	-0.09	-0.19	-0.11	-0.28	-0.20	-0.05
KGE	0.42	0.61	0.65	0.87	0.76	0.88	0.47	0.79	0.71

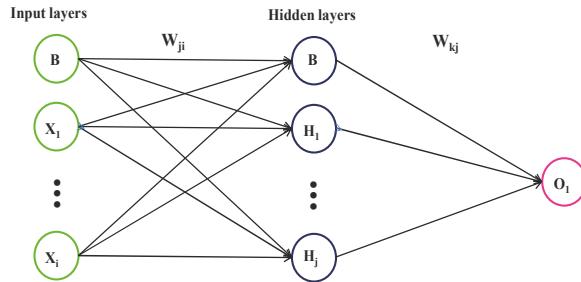
### 3.3. Bias Correction of Real-Time Forecasts

The NWP models approximate mathematically the physical dynamics using nonlinear differential equations; however, these approximations include uncertainties due to the complex system of the atmosphere [48]. Advancements in meteorological forecast models did not solve the issues related to the inevitable biases. Despite the efforts to incorporate all sources of uncertainty into the forecast and the methodologies applied to generate the forecast ensembles, the results are still subject to errors and systematic biases [49]. Statistical postprocessing increases the accuracy of the forecast data by decreasing the errors. ANN, as a postprocessing method, can determine the complex relationships between the inputs and outputs. ANN has been widely used in the hydrology and modeling of water resource systems [5,12–29].

#### 3.3.1. Description of ANN

ANN is biologically inspired from neurons in the brain and consists of the interaction of computational units. ANN establishes a relationship between the input and target and produces a correct response by following the processes of human brain activities such as saving information, learning, and training [5]. The structure of an ANN includes the input layer (including the input nodes connected to the input variables), hidden layer (it can be one hidden layer or more layers which include the hidden nodes) and output layer (including the output nodes that deliver the output data). Different weights are connected to the nodes (units) of the layers in different forms such as the Feedforward Neural Network (FNN), Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN). A three-layer FNN is illustrated in Figure 3. Here,  $i$ ,  $j$  and  $k$  are the number of nodes for the input, hidden and output layers, respectively. Furthermore,  $X_i$  is the input variable,  $O_j$  is the output variable,  $W_{ji}$  is the connected weight from the input layer to the hidden layer, and  $W_{kj}$  is the connected weight from the hidden layer to the

output layer. In this study, the FNN is applied to construct the neural network. The weights and bias nodes in the feedforward network are connected in one forward direction.



**Figure 3.** An example schematic of a feedforward network.

ANN is characterized by the simulation and response processes to learn the correct response for each input through training. The general function between input and output is called a transfer (activation) function, which can be a linear or nonlinear function. In this study, the nonlinear sigmoid function is applied for the neural networks. The transfer function converts the summation of the weights ( $w$ ) and inputs ( $x$ ) and ( $b$ ) bias to an output vector by the following equation:

$$y = f(\sum w \times x + b) \quad (16)$$

The inputs are multiplied by weights and then fed to the first hidden layer. The weights can be chosen by minimizing the following error function:

$$E = \frac{1}{2} [Tar_j - O_j(w, x)]^2 \quad (17)$$

where  $O_j$  is the output of the output layer and  $Tar_j$  is the corresponding target. The procedure of input weight adjustment is called training (learning). In this study, The Levenberg-Marquardt (trainlm) was chosen as the training function and the back propagation generalized delta rule (BPGDR), is applied as the training algorithm that minimizes the error function based on the differences between the modeled and desired outputs. The BPGDR formulation can be summarized in two parts as follows:

1. is for the output weights, which are connected to the output nodes:

$$\Delta w_{kj} = \eta \frac{\partial E}{\partial w_{kj}} \quad (18)$$

2. is for the hidden weights, which are connected to the hidden nodes:

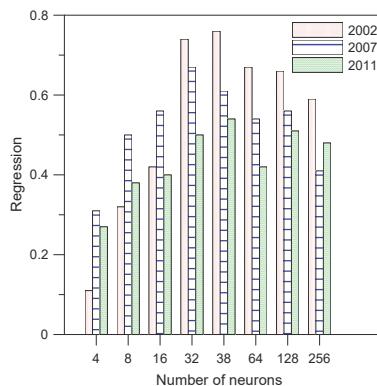
$$\Delta w_{ji} = \eta \frac{\partial E}{\partial w_{ji}} \quad (19)$$

where  $w_{kj}$  and the  $w_{ji}$  are the output and hidden weights, respectively,  $E$  is the error function, and  $\eta$  is the learning rate.

### 3.3.2. Application of ANN for Real-Time Bias Correction

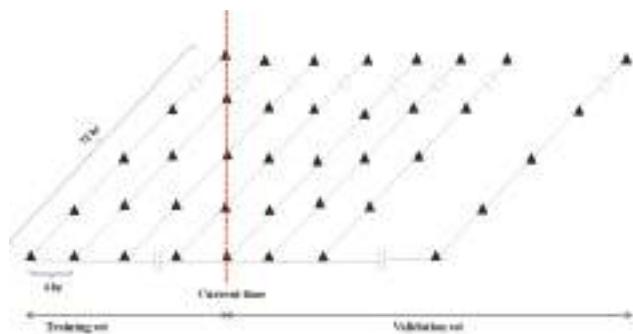
ANN learns the error structure from historical data and corresponding observations. Then, the trained network can be used to reproduce bias-corrected predictions. To apply bias correction in a real-time forecast system, the current time is selected, and the training data set is considered

until the current time. The remaining data are then considered as the validation data set. In the training set, the observation is the target, and the forecast is the input data. The objective of using ANN is to find a nonlinear relationship between the input and the target data to achieve the minimum error between the input and target data. A network with few neurons restricts the network learning ability while one with many neurons may lead to overfitting and poor generalization of the network. Different numbers of the neurons used to construct the model and results showed that by increasing the number of neurons the model performance improved; after 32 neurons the results did not dramatically improved (Figure 4). There are 38 sub-basins in the study area, therefore the 38 neurons were chosen. The stopping criterion used in this study is early stopping; this avoids the overtraining and overfitting of the model for the training data set.



**Figure 4.** The results of the regression for different number of the neurons.

The schematic of choosing the training and validation data for the ANN construction in this study is provided in Figure 5.



**Figure 5.** The schematic of the selection of training and validation set data in the real-time forecast of the WRF model output data.

The evaluation criteria used in this study include the Relative Bias (RB), Mean Relative Error (MRE), and the Mean Absolute Error (MAER), as given in Table 2, NSE and REV. In the formulas,  $O_i$  is the observed data, and  $F_i$  is the forecast data. The RB metric is selected to compare the forecast and observed values by calculating the absolute bias which is normalized by the sum of the observed values. The MRE was also selected to compare the simulated and observed values, the MRE shows the average of the relative error which is the uncertainty of the measurement compared to the measurement.

To show the average of the absolute differences between observed and forecast data the MAER was selected to have a comparison between the simulated and observed values.

**Table 2.** Statistical measures used to evaluate the performance.

Index	Formula
Relative Bias (RB)	$RB = \frac{\sum_{i=1}^n (O_i - f_i)}{\sum_{i=1}^n O_i}$
Mean Relative Error (MRE)	$MRE = \frac{1}{n} \sum_{i=1}^n \frac{ S_i - O_i }{O_i}$
Mean Absolute Error (MAER)	$MAER = \frac{1}{n} \sum_{i=1}^n  O_i - f_i $

#### 4. Results

##### 4.1. Real-Time Accuracy Improvement of the Precipitation

In real-time flood forecasting, it is undeniable that the accuracy of the real-time precipitation plays the most important role in achieving accurate real-time flood forecasts. In this study, ANN is applied for the bias correction of the real-time precipitation produced by the WRF model. A comparison of the observed rainfall and with/without bias-adjusted real-time WRF rainfall indicated that the accuracy improved with the ANN bias correction (Table 3). The commonly used statistical error measurements such as MRE, RB, REV and MAER compared the skill of model forecasts and observations for error measurement. The results showed that the ANN real-time bias correction improved all statistical terms for the forecast rainfall. A comparison of the results indicated that the use of the bias correction method improved the real-time forecast error measurement statistics by 70.09, 81.61 and 70.49% for the 2002, 2007 and 2011 events, respectively.

**Table 3.** Results of the improvements to the flood events in 2002, 2007 and 2011.

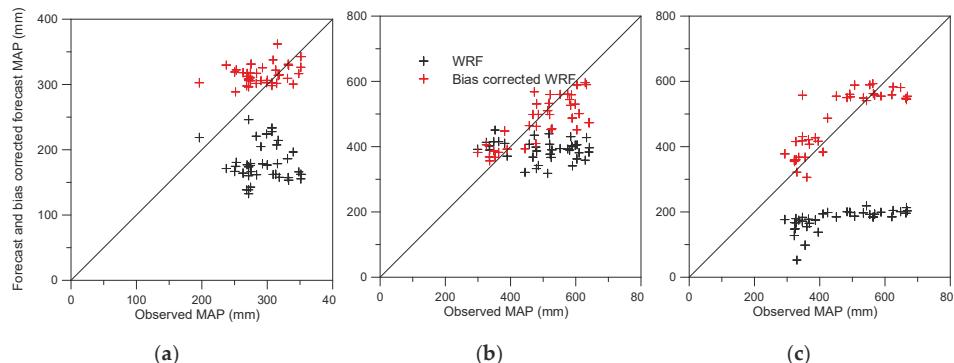
Event	Forecast data	MRE	RB	REV	MAER
2002	WRF	38.45	67.73	38.78	114.47
	Bias-adjusted WRF	11.64	42.91	7.74	31.17
	Improvement (%)	69.71	57.84	80.05	72.77
2007	WRF	42.40	35.42	23.20	105.84
	Bias-adjusted WRF	10.55	4.03	4.13	20.67
	Improvement (%)	75.12	88.62	82.22	80.47
2011	WRF	65.61	85.81	61.54	59.83
	Bias-adjusted WRF	27.24	13.75	14.82	21.55
	Improvement (%)	58.09	83.98	75.92	63.98

Further investigations are performed to assess the sum, minimum, maximum and percentage of the underestimation of the observed, forecast and bias-corrected forecast rainfall for the 2002, 2007 and 2011 events. Table 4 shows the results of the MAP analysis for the flood events before and after bias correction using ANN. The results obtained before and after real-time ANN bias correction led to an accuracy improvement of MAP by increasing the underestimation and the skill of real-time forecast WRF data. According to the results, after applying the ANN bias correction, the underestimation improved by 65.79, 23.69 and 73.68% for the 2002, 2007 and 2011 events, respectively.

**Table 4.** Statistics of the MAP for the 2002, 2007 and 2011 flood events.

Event	Data	$\Sigma$ (mm)	Min (mm)	Max (mm)	Underestimation (%)	Error Reduction (%)
2002	Observation	11100.52	196.19	351.40	-	-
	WRF	6795.70	132.64	246.12	97.37	-
	WRF-revised	11959.45	288.62	362.03	31.58	75.28
2007	Observation	1904.96	299.06	641.59	-	-
	WRF	14622.80	158.51	450.84	78.95	-
	WRF-revised	18255.56	356.49	596.11	55.26	89.53
2011	Observation	17445.92	293.18	743.25	-	-
	WRF	6709.83	52.56	218.64	100	-
	WRF-revised	18021.18	306.21	593.00	34.21	88.74

In the present study, a comparison between the observed and forecast accumulated MAP in all sub-basins is performed to compare the accuracy before and after bias correction. It is evident from Figure 6 that after bias correction, the accumulated WRF data have improved significantly. The scatterplot of the observed, real-time, and bias-corrected real-time forecast MAP indicated that the ANN bias correction method applied to the Imjin basin increased the real-time forecast accuracy by decreasing the underestimation of the precipitation in all the sub-basins. A comparison of the results of the accumulated observed rainfall and the real-time forecast WRF data showed that the ANN bias correction method used in this study was able to reduce the biases of real-time precipitation with the desired accuracy.



**Figure 6.** The scatterplot of the accumulated observed MAP and mean ensemble of real-time WRF data before and after bias correction for the events in (a) 2002, (b) 2007 and (c) 2011.

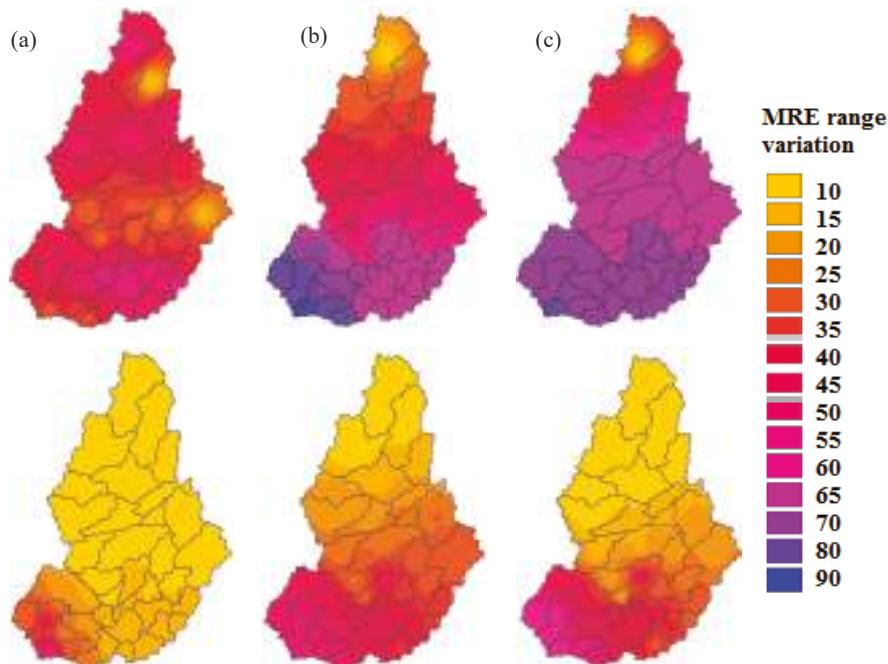
The spatial distribution of the MRE on the catchment scale indicated that the ANN bias correction led to a decrease of the MAER in all the sub-basins (Figure 7). The results of the MRE assessment of the 38 sub-basins indicated that the error reduction of the forecast MAP varies by sub-basin in the study area. The results of the ANN real-time bias correction showed that the average improvements of the MRE over the catchment are 69.71, 61.24 and 53.90% for the 2002, 2007 and 2011 events, respectively. By applying the forecast bias correction, the forecast capability and the accuracy of the model predictions improved remarkably.

#### 4.2. Real-Time Flood Forecasting Accuracy Improvement

In this section, the real-time flood forecast accuracy evaluation is performed by comparing the observed, simulated and forecasted values. The simulation floods were estimated using the observation MAP as input to the SURR model, while the forecast was obtained using the real-time WRF data as the input to the SURR model. The statistical error measurements (NSE, KGE, MRE and REV) were used to

compare the performance of hydrologic simulations and forecasts before and after bias correction and determine the average deviations.

In this study, we used gauge and WRF precipitation data to drive the hydrological model to simulate and forecast flood events. Here, the 72-h real-time forecast data at a spatial resolution of  $1\text{ km} \times 1\text{ km}$  and a temporal resolution of 10 min are used as input to the SURR model over the Imjin basin; furthermore, the bias-corrected WRF data are used as the input data for the SURR model. The coupling system of the SURR and WRF model includes the use of the observed precipitation until the start of the forecast time, and then continues to apply the WRF data for a 72-h forecast lead time. The observed and real-time WRF data (bias-corrected and raw data) are used to run the hydrological model, and this process is repeated for the next 6 h to the end of the forecast time. Among the various forecast ensembles, only one stream flow forecast is shown with the relevant precipitation to indicate the real-time forecast discharge variation over time due to space limitations. According to the findings of this study, by applying the bias-adjusted WRF data, the accuracy improvement was suggested by the increasing NSE and KGE and decreasing MRE and REV (Table 5).



**Figure 7.** The comparison of the spatial distribution of the MRE in 38 sub-basins before (upper part) and after bias correction (down part) for the events in (a) 2002, (b) 2007 and (c) 2011.

The performance of the SURR model in simulating the streamflow along with the SURR-WRF coupled model before and after bias correction in forecasting the streamflow at the Gunnam, Jeonkuk and Jeogseong stations in the 2002, 2007 and 2011 events are illustrated in Figure 8. The observed stream flow (black dashed curve), the forecast stream flow using WRF data (blue curve), the forecast stream flow using bias-corrected WRF data (green curve) and the simulated streamflow (red curve) are presented in Figure 8. The Jeonkuk station has no observation data for the 2002 event.

**Table 5.** Comparison of results for the statistical error measurement for the 2002, 2007 and 2011 events.

Index	Station	SURR	SURR-WRF	SURR-Revised WRF	Improvement (%)
Event 2002					
NSE	Gunnam	0.26	-18.27	-7.24	60.21
MRE		-0.09	-0.95	-0.24	74.74
REV		0.16	0.70	0.43	38.57
KGE		0.41	-1.20	-0.52	56.67
NSE	Jeogseong	0.68	-19.85	-8.68	56.27
MRE		-0.25	0.80	0.26	67.50
REV		0.03	0.53	0.27	49.06
KGE		0.60	-1.14	-0.68	40.35
Event 2007					
NSE	Gunnam	0.69	-4.57	-2.01	56.02
MRE		-0.58	-0.60	-0.56	6.67
REV		-0.48	-0.57	-0.52	8.77
KGE		0.53	-5.03	-3.98	20.87
NSE	Jeonkuk	0.78	-6.63	-0.82	87.63
MRE		-0.60	-0.77	-0.37	51.95
REV		-0.12	-0.22	-0.18	18.18
KGE		0.62	-2.77	-1.65	40.43
NSE	Jeogseong	0.71	-10.30	-5.71	44.56
MRE		-0.69	-0.78	-0.65	16.67
REV		-0.52	-0.54	-0.59	9.26
KGE		0.51	-3.30	-2.24	32.12
Event 2011					
NSE	Gunnam	0.80	-0.47	0.07	85.11
MRE		-0.49	-0.79	-0.51	35.44
REV		-0.08	-0.59	-0.37	37.29
KGE		0.81	-0.26	-0.09	65.38
NSE	Jeonkuk	0.81	-0.87	-0.06	93.10
MRE		-0.63	-0.67	-0.58	13.43
REV		-0.34	-0.73	-0.42	42.47
KGE		0.60	-0.79	-0.21	73.42
NSE	Jeogseong	0.90	-1.06	-0.23	78.30
MRE		-0.06	-0.56	-0.07	87.50
REV		-0.45	-0.60	-0.32	46.67
KGE		0.81	-1.22	-0.46	62.29

The observed and real-time forecast precipitation (with/without bias correction) events are shown separately in different panels. Comparing the flood observation and simulation, the rising and falling limb and amplitude of the simulated flood, which used observed precipitation, is quite similar to the observations, while the forecasted flood by the WRF model followed similar trend by significant underestimation. This result can be explained by the fact that observation and real-time forecast precipitation are two various sources of input data that are forced into the SURR model.

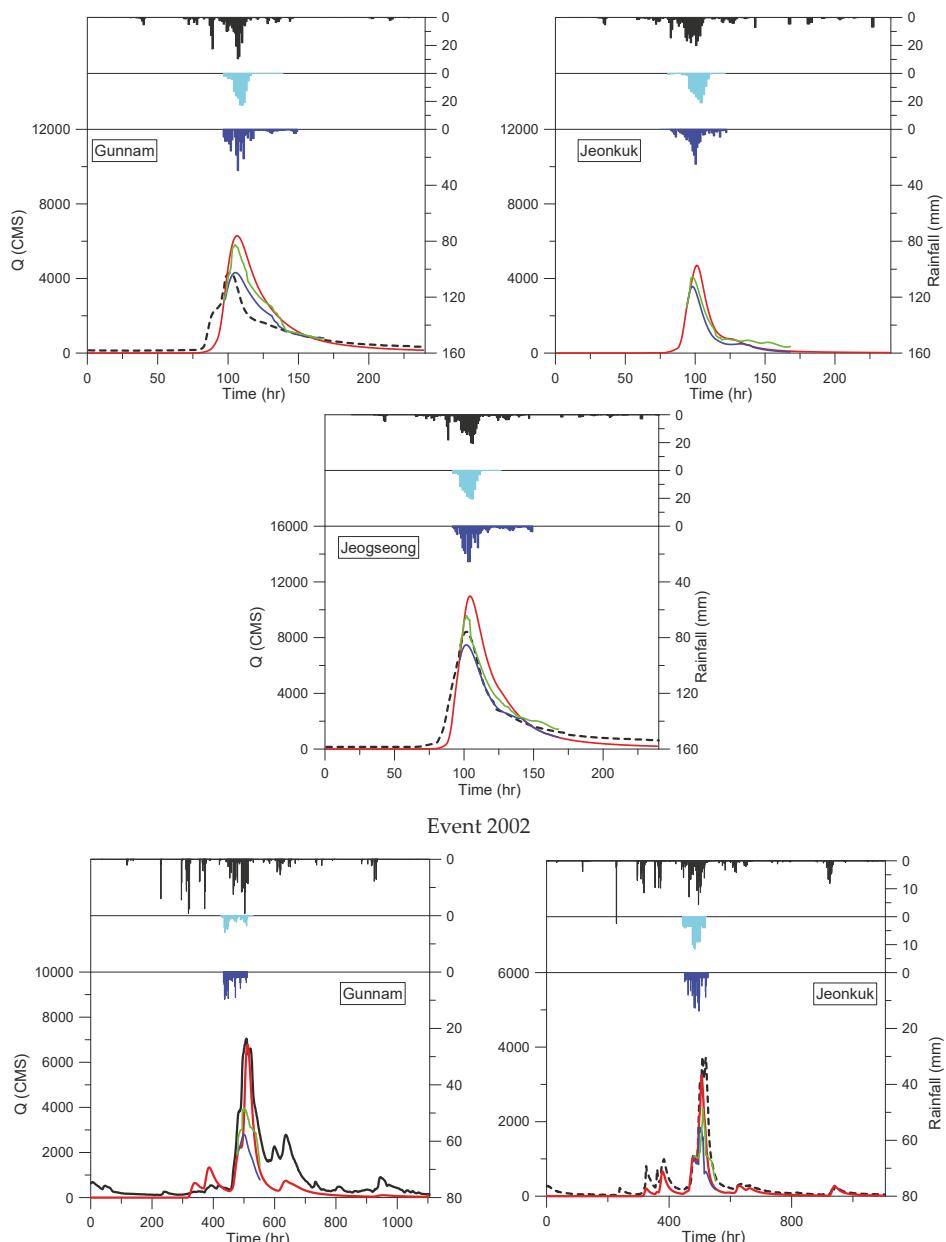
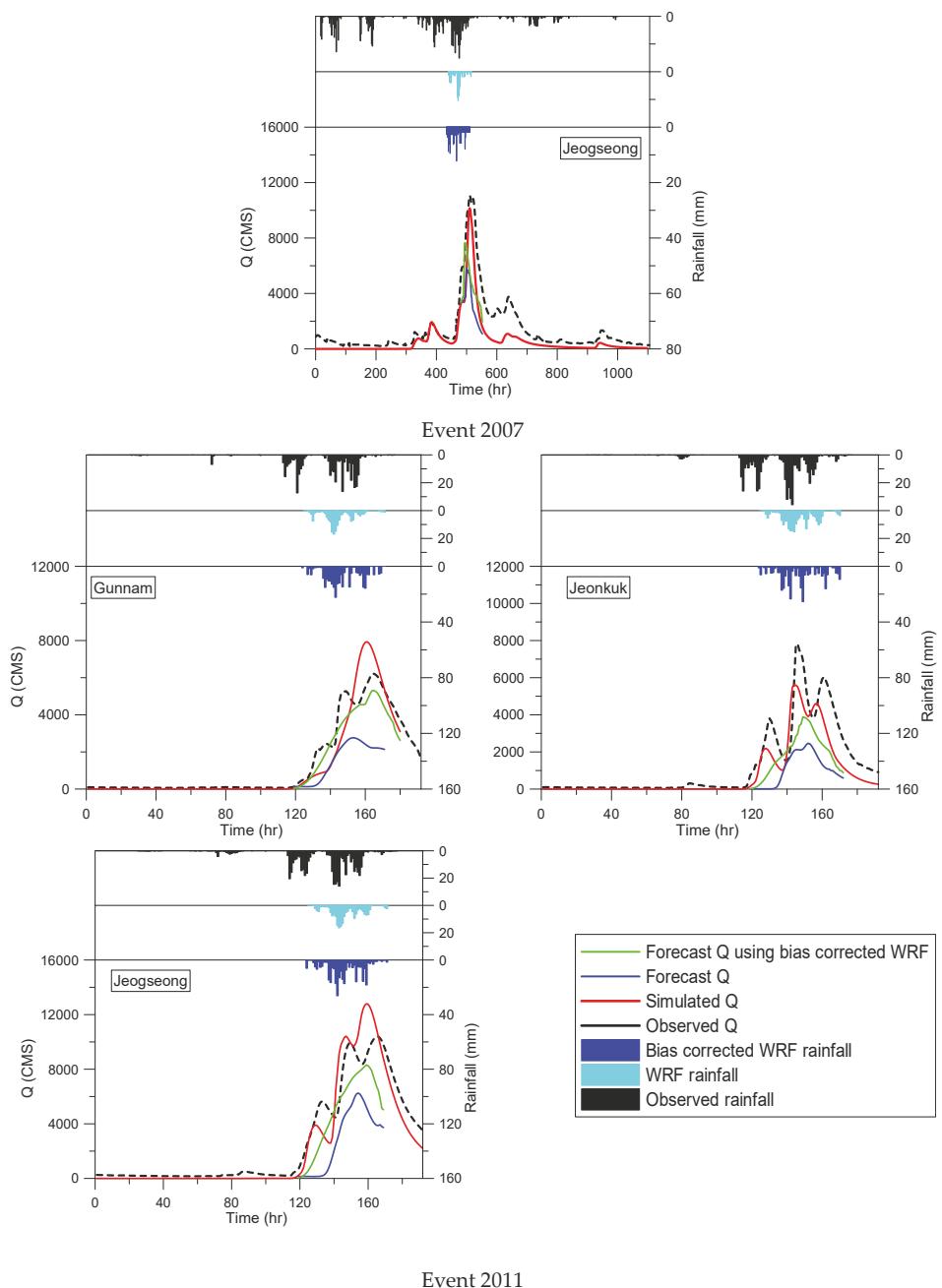


Figure 8. Cont.



**Figure 8.** Comparison of simulated, observed and forecasted flows (before and after bias correction) and observed and forecasted rainfall (before and after bias correction) for 2002, 2007 and 2011 events.

## 5. Discussion

The NWP models determine the future state of the weather by forcing current weather conditions into the atmospheric models. The chaotic and nonlinear system of the atmosphere and complex nature of precipitation made precipitation difficult to forecast accurately with the NWP models. In general, the NWP models forecast less accurate precipitation rates than indicated by the observed precipitation rates. The precipitation intensity [50] and location [51] are challenging for precipitation forecasting. The initial and boundary conditions such as horizontal resolution, domain size as grid spacing and physical parameterization schemes are directly related to the NWP model results in heavy precipitation forecasting [37,52]. The Quantitative Precipitation Forecast (QPF) accuracy in three-dimensional primitive-equation atmosphere circulation models, such as WRF, can be influenced by different initializations, microphysics, and planetary boundary layer (PBL) schemes [53]. In this study, the WRF real-time forecast data, in comparison with the observed data, had systematic biases; the errors related to the NWP model forecasts were reflected in the underestimation of the real-time precipitation forecast by the WRF model. The cumulative MAP had the underestimations of 97.37, 78.95 and 100% for the 2002, 2007 and 2011 events, respectively. Because the Imjin River is a transboundary river, a reliable streamflow forecast is needed for this watershed. Improving the streamflow forecast in the Imjin basin is a highly important task especially during heavy rainfall and extreme events. For a reliable flood forecast, an accurate forecast of rainfall is needed, but the use of raw WRF data caused large biases in the flood forecast.

The aim of the present study is to improve real-time flood forecasting by applying ANN as a postprocessing technique. Considering all real-time forecast cases, it can be concluded that on average, hydrological forecasts based on the WRF model forecast reproduced stream flow with a significant underestimation. ANN was able to construct a relationship between the input and output data to reduce the biases between the observed and forecast rainfall data. Therefore, the use of the ANN bias correction resulted in the improvement of the real-time flood forecasts in the Imjin basin for the Gunnam, Jeonkuk and Jeogseong stations. By applying the ANN bias correction, errors in precipitation forecasts are modified, which resulted in the real-time precipitation forecast accuracy enhancement, which is not uniform over the forecast interval. The results indicated that there was significant improvement in the statistical errors in the forecast MAP before and after applying the bias correction method. By using the ANN bias correction, the underestimation of the real-time forecast data and the accumulated forecast MAP improved by 65.57, 30.03 and 65.79% for the 2002, 2007 and 2011 events, respectively. The use of the bias-corrected precipitation resulted in the significant improvement in the real-time flood forecasting by 57.53, 33.31 and 57.70% for the 2002, 2007 and 2011 events, respectively. The promising results indicated that the ANN bias correction in the Imjin River had resulted in the improved performance of real-time flood forecasting.

The forecast verification indicated a noteworthy increase in forecast performance with ANN, compared to the raw model outputs. The precipitation real-time forecast accuracy enhancement can translate to the improvement in real-time flood forecasts. However, the results of the real-time flood forecasts were still underestimated because a perfect estimate of real-time forecast rainfall quantity is not easily obtained. This complexity is the result of the two sources of precipitation, which are quite different. The spatial and temporal variation in the rainfall characteristics were not captured well by the WRF model in the real-time forecast data. Typically, the forecasted floods underestimated the peak floods, and the forecasted flood errors are related to the inaccuracies in the real-time forecasted rainfall. Considering all real-time forecast cases from the start of the forecasting time until the end of the forecasting time, on average, it can be concluded that hydrological forecasts based on meteorological model inputs were able to reproduce the shape and the timing of the calculated stream flow fairly well. However, the underestimation of the WRF model bias-corrected precipitation was affected by the real-time forecast discharge in all events. Another reason for the uncertainties in real-time flood forecasting after bias correction could be related to the fact that the hydrological model calibration was performed with rain gauge data and observed streamflow data; however, the coupled SURR-WRF

model used the real-time forecasted rainfall. These different sources of rainfall were used as inputs for the hydrological models. Therefore, it could be expected that the hydrological response in forecasting the streamflow would not match the simulated streamflow very well. For the real-time flood forecasts, although the real-time flood forecasts are not perfectly matched the observed values even after applying the bias correction, available forecast data are preferred over completely ignoring future events of interest. Due to the importance of the Imjin basin for both North and South Korea, available information regarding flood forecasts in the studied area is valuable.

## 6. Conclusions and Recommendations

In hydrometeorological studies, the performance of coupled hydrometeorological models is directly dependent on the accuracy of the forecasted precipitation. Because the NWP models cannot forecast precipitation accurately, postprocessing of the output of NWP models is necessary. The postprocessing of the real-time systems can be performed using historical data and forecasts by ANN bias correction. In this study, ANN is applied for the bias correction of the real-time forecast of the WRF model. The bias correction is estimated to quantify the accuracy improvement of the rainfall and corresponding flood forecasts in a transboundary river. The main conclusions of this study are summarized below:

- (1) Applying ANN for bias correction improved the forecast performance by reducing MRE, RB, REV and MAER by 69.71, 57.84, 80.05 and 72.77%, respectively, in the 2002 event; by 75.12, 88.62, 82.22 and 80.47%, respectively, in the 2007 event; and by 58.09, 83.98, 75.92 and 63.98%, respectively, in the 2011 event.
- (2) The sum, minimum, maximum and the underestimation of the WRF real-time forecast data were improved after applying the ANN bias correction to the real-time WRF data.
- (3) By applying the ANN bias correction, the underestimation of WRF data improved 65.79, 23.69 and 73.68% in the 2002, 2007 and 2011 events, respectively. The error was also reduced by 75.28, 89.53 and 88.74% over the Imjin catchment in terms of the accumulated MAP in the 2002, 2007 and 2011 events, respectively.
- (4) The error comparison in each sub-basin indicated that the average percentage of MRE reduction in the catchment was 69.71, 61.24 and 53.90% for the 2002, 2007 and 2011 events, respectively.
- (5) By applying the ANN bias correction, the performance of the SURR-WRF coupled models in real-time flood forecasts increased by increasing the NSE and KGE and reducing the MRE and REV for Gunnam, Jeonkuk and Jeogseong stations.

Additional analyses could be performed to compare the benefits of using different QPF postprocessing techniques such as radar data assimilation, regression methods and Kalman filtering.

**Author Contributions:** Conceptualization, J.A. and B.D-H.; Data curation, J.A.; Formal analysis, J.A.; Funding acquisition, B.D-H.; Methodology, A.J.; Project administration, B.D-H.; Resources, J.A. and B.D-H.; Supervision, B.D-H.; Validation, J.A.; Visualization, J.A. and B.D-H.; Writing—original draft, J.A.; Writing—review & editing, J.A. and B.D-H.

**Funding:** This research was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure, and Transport (18AWMP-B083066-05) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2011-0030040).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Liu, J.; Wang, J.; Pan, S.; Tang, K.; Li, C.; Han, D. A real-time flood forecasting system with dual updating of the NWP rainfall and the river flow. *Nat. Hazards* **2015**, *77*, 1161–1182. [[CrossRef](#)]

2. Vincendon, B.; Ducrocq, V.; Nuissier, O.; Vie', B. Perturbation of convection-permitting NWP forecasts for flash-flood ensemble forecasting. *Nat. Hazards Earth Syst. Sci.* **2011**, *11*, 1529–1544. [[CrossRef](#)]
3. Lin, C.A.; Wen, L.; Lu, G.; Wu, Z.; Zhang, J.; Yang, Y.; Zhu, Y.; Tong, L. Real-time forecast of the 2005 and 2007 summer severe floods in the Huaihe River Basin of China. *J. Hydrol.* **2010**, *381*, 33–41. [[CrossRef](#)]
4. Davolio, S.; Miglietta, M.M.; Diomede, T.; Marsigli, C.; Montani, A. A flood episode in northern Italy: Multi-model and single-model mesoscale meteorological ensembles for hydrological predictions. *Hydrol. Earth Syst. Sci.* **2013**, *17*, 2107–2120. [[CrossRef](#)]
5. Moghim, S.; Bras, R.L. Bias Correction of Climate Modeled Temperature and Precipitation Using Artificial Neural Networks. *J. Hydrometeorol.* **2017**, *18*, 1867–1884. [[CrossRef](#)]
6. Yang, Z.; Hsu, K.; Sorooshian, S.; Xu, X.; Braithwaite, D.; Verbist, K.M.J. Bias Adjustment of Satellite-based Precipitation Estimation using Gauge Observations-A Case Study in Chile. *J. Geophys. Res.-Atmos.* **2016**, *121*, 3790–3806. [[CrossRef](#)]
7. Crochemore, L.; Ramos, M.H.; Pappenberger, F. Bias correcting precipitation forecasts to improve the skill of seasonal streamflow forecasts. *Hydrol. Earth Syst. Sci.* **2016**, *20*, 3601–3618. [[CrossRef](#)]
8. Rogelis, M.C.; Werner, M. Streamflow forecasts from WRF precipitation for flood early warning in mountain tropical areas. *Hydrol. Earth Syst. Sci.* **2017**, *22*, 853–870. [[CrossRef](#)]
9. Applequist, S.; Garhs, G.; Pfeffer, R. Comparisons of methodologies for probabilistic quantitative precipitation forecasting. *Weather Forecast.* **2002**, *17*, 783–799. [[CrossRef](#)]
10. Charba, J.; Samplatsky, F. High-resolution GFS-based MOS quantitative precipitation forecasts on a 4-km grid. *Mon. Weather Rev.* **2011**, *139*, 39–68. [[CrossRef](#)]
11. Cane, D.; Ghigo, S.; Rabuffetti, D.; Milelli, D. Real-time flood forecasting coupling different post processing techniques of precipitation forecast ensembles with a distributed hydrological model. The case study of May 2008 flood in western Piemonte, Italy. *Nat. Hazards Earth Syst. Sci.* **2013**, *13*, 211–220. [[CrossRef](#)]
12. Chen, J.; Brissette, F.P.; Chaumont, D.; Braun, M. Finding appropriate bias correction methods in downscaling precipitation for hydrologic impact studies over North America. *Water Resour. Res.* **2013**, *49*, 4187–4205. [[CrossRef](#)]
13. Tapiador, F.J.; Kidd, C.; Hsu, K.-L.; Marzano, F.S. Neural networks in satellite rainfall estimation. *Meteorol. Appl.* **2004**, *11*, 83–91. [[CrossRef](#)]
14. Hsu, K.; Gao, H.; Sorooshian, S.; Gupta, H. Precipitation estimation from remotely sensed information using artificial neural networks. *J. Appl. Meteorol. Climatol.* **1997**, *36*, 1176–1190. [[CrossRef](#)]
15. Rivolta, G.; Marzano, F.S.; Coppola, E.; Verdecchia, M. Artificial neural-network technique for precipitation nowcasting from satellite imagery. *Adv. Geosci.* **2006**, *7*, 97–103. [[CrossRef](#)]
16. Aichouri, I.; Hani, A.; Bougerira, N.; Djabri, L.; Chaffai, H.; Lallaheem, S. River flow model using artificial neural networks. *Energy Proced.* **2015**, *74*, 1007–1014. [[CrossRef](#)]
17. Wu, C.L.; Chau, K.W. Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis. *J. Hydrol.* **2011**, *399*, 394–409. [[CrossRef](#)]
18. Peng, T.; Zhou, J.; Zhang, C.; Fu, W. Streamflow forecasting using empirical wavelet transform and artificial neural networks. *Water* **2017**, *9*, 406. [[CrossRef](#)]
19. Wang, Y.; Min, J.; Chen, Y.; Xiang-Yu, H.; Zeng, M.; Li, X. Improving precipitation forecast with hybrid 3DVar and time-lagged ensembles in a heavy rainfall event. *Atmos. Res.* **2017**, *183*, 1–16. [[CrossRef](#)]
20. Chiang, Y.M.; Chang, F.J. Integrating hydrometeorological information for rainfall-runoff modelling by artificial neural networks. *Hydrol. Process.* **2009**, *23*, 1650–1659. [[CrossRef](#)]
21. Ahmad, S.; Simonovic, S.P. An artificial neural network model for generating hydrograph from hydro-meteorological parameters. *J. Hydrol.* **2005**, *315*, 236–251. [[CrossRef](#)]
22. Chang, F.J.; Tsai, M.J. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *J. Hydrol.* **2016**, *535*, 256–269. [[CrossRef](#)]
23. Tayfur, G.; Singh, V.P.; Moramarco, T.; Barbetta, S. Flood Hydrograph Prediction Using Machine Learning Methods. *Water* **2018**, *10*, 968. [[CrossRef](#)]
24. Zhou, J.; Peng, T.; Zhang, C.; Sun, N. Data Pre-Analysis and Ensemble of Various Artificial Neural Networks for Monthly Streamflow Forecasting. *Water* **2018**, *10*, 628. [[CrossRef](#)]
25. Chang, L.-C.; Amin, M.Z.M.; Yang, S.-N.; Chang, F.-J. Building ANN-Based Regional Multi-Step-Ahead Flood Inundation Forecast Models. *Water* **2018**, *10*, 1283. [[CrossRef](#)]

26. Chang, L.C.; Shen, H.Y.; Chang, F.J. Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. *J. Hydrol.* **2014**, *519*, 476–489. [[CrossRef](#)]
27. Tien Bui, D.; Khosravi, K.; Li, S.; Shahabi, H.; Panahi, M.; Singh, V.P.; Chapi, K.; Shirzadi, A.; Panahi, S.; Chen, W.; et al. New Hybrids of ANFIS with Several Optimization Algorithms for Flood Susceptibility Modeling. *Water* **2018**, *10*, 1210. [[CrossRef](#)]
28. Maier, H.R.; Dandy, G.C. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environ. Model. Softw.* **2000**, *15*, 101–124. [[CrossRef](#)]
29. Maier, H.R.; Jain, A.; Dandy, G.C.; Sudheer, K.P. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environ. Model. Softw.* **2010**, *25*, 891–909. [[CrossRef](#)]
30. Singh, P.; Borah, B. Indian summer monsoon rainfall prediction using artificial neural network. *Stoch. Environ. Res. Risk Assess.* **2013**, *27*, 1585–1599. [[CrossRef](#)]
31. Maier, H.R.; Kapelan, Z.; Kasprzyk, J.; Kollat, J.; Matott, L.S.; Cunha, M.C.; Dandy, G.C.; Gibbs, M.S.; Keedwell, E.; Marchi, A.; et al. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environ. Model. Softw.* **2014**, *62*, 271–299. [[CrossRef](#)]
32. Lee, S.; Bae, D.H.; Cho, C.H. Changes in future precipitation over South Korea using a global high-resolution climate model. *Asia-Pac. J. Atmos. Sci.* **2013**, *49*, 619–624. [[CrossRef](#)]
33. Lee, J.; Choi, J.; Lee, O.; Yoon, J.; Kim, S. Estimation of Probable Maximum Precipitation in Korea using a Regional Climate Model. *Water* **2017**, *9*, 240. [[CrossRef](#)]
34. Kim, N.W.; Won, Y.S.; Chung, I.M. The scale of typhoon RUSA. *Hydrol. Earth Syst. Sci.* **2006**, *3*, 3147–3182. [[CrossRef](#)]
35. You, C.H.; Lee, D.I.; Jang, S.M.; Jang, M.; Uyeda, U.; Shinoda, Y.; Kobayashi, F. Characteristics of rainfall systems accompanied with Changma front at Chujado in Korea. *ASIA-PAC. J. Atmos. Sci.* **2010**, *46*, 41–51. [[CrossRef](#)]
36. Kim, E.; Ra, I.; Rhee, K.H.; Kim, C.S. Estimation of real-time flood risk on roads based on rainfall calculated by the revised method of missing rainfall. *Sustainability* **2014**, *6*, 6418–6431. [[CrossRef](#)]
37. Jee, J.-B.; Kim, S. Sensitivity Study on High-Resolution WRF Precipitation Forecast for a Heavy Rainfall Event. *Atmosphere* **2017**, *8*, 96. [[CrossRef](#)]
38. Yucel, I.; Onen, A.; Yilmaz, K.; Gochis, D. Calibration and evaluation of a flood forecasting system: Utility of numerical weather prediction model, data assimilation and satellite-based rainfall. *J. Hydrol.* **2015**, *523*, 49–66. [[CrossRef](#)]
39. Lee, D.K.; Eom, D.Y.; Kim, J.W.; Lee, J.B. High-resolution summer rainfall prediction in the JHWC real-time WRF system. *Asia-Pac. J. Atmos. Sci.* **2010**, *46*, 341–353. [[CrossRef](#)]
40. Srivastava, P.K.; Han, D.; Rico-Ramirez, M.A.; O'Neill, P.; Islam, T.; Gupta, M.; Dai, Q. Performance evaluation of WRF-Noah Land surface model estimated soil moisture for hydrological application: Synergistic evaluation using SMOS retrieved soil moisture. *J. Hydrol.* **2015**, *529*, 200–212. [[CrossRef](#)]
41. Skamarock, W.C.; Klemp, J.B.; Dudhia, J.; Gill, D.O.; Barker, D.M.; Duda, G.; Huang, X.; Wang, W.; Powers, J.G. *A Description of the Advanced Research WRF Version 3*; Tech. Note, NCAR/TN-475+STR; National Center for Atmospheric Research: Boulder, CO, USA, 2008.
42. Bae, D.H.; Lee, B.J. Development of Continuous Rainfall Runoff Model for Flood Forecasting on the Large Scale Basin. *J. Korea Water Resour. Assoc.* **2011**, *44*, 51–64. [[CrossRef](#)]
43. Kimura, T. *The Flood Runoff Analysis Method by the Storage Function Model*; The Public Works Research Institute Ministry of Construction: Tsukuba, Japan, 1961.
44. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through. Part. I. A conceptual model discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [[CrossRef](#)]
45. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* **2009**, *377*, 80–91. [[CrossRef](#)]
46. Seo, B.C.; Quintero, F.; Krajewski, F.W. High-Resolution QPF Uncertainty and Its Implications for Flood Prediction: A Case Study for the Eastern Iowa Flood of 2016. *J. Hydrometeor.* **2016**, *19*, 1289–1304. [[CrossRef](#)]
47. Krause, P.; Boyle, D.P.; Bäse, F. Comparison of different efficiency criteria for hydrological model assessment. *Adv. Geosci.* **2005**, *5*, 89–97. [[CrossRef](#)]
48. Cloke, H.L.; Pappenberger, F. Ensemble flood forecasting: A review. *J. Hydrol.* **2009**, *375*, 613–626. [[CrossRef](#)]

49. Madadgar, S.; Moradkhani, H.; Garen, D. Towards improved post-processing of hydrologic forecast ensembles. *Hydrol. Process.* **2012**, *28*, 104–122. [[CrossRef](#)]
50. Kobold, M.; Suselj, K. Precipitation forecasts and their uncertainty as input into hydrological models. *Hydrol. Earth Syst. Sci.* **2005**, *9*, 322–332. [[CrossRef](#)]
51. Ebert, E.; McBride, J. Verification of precipitation in weather systems: Determination of systematic errors. *J. Hydrol.* **2000**, *239*, 179–202. [[CrossRef](#)]
52. Wang, X.; Steinle, P.; Seed, A.; Xiao, Y. The Sensitivity of Heavy Precipitation to Horizontal Resolution, Domain Size, and Rain Rate Assimilation: Case Studies with a Convection-Permitting Model. *Adv. Meteorol.* **2016**, *2016*, 7943845. [[CrossRef](#)]
53. Jankov, I.; Gallus, W.A.; Segal, M.; Koch, S.E. Influence of Initial Conditions on the WRF–ARW Model QPF Response to Physical Parameterization Changes. *Weather Forecast.* **2007**, *22*, 501–519. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

MDPI  
St. Alban-Anlage 66  
4052 Basel  
Switzerland  
Tel. +41 61 683 77 34  
Fax +41 61 302 89 18  
[www.mdpi.com](http://www.mdpi.com)

*Water* Editorial Office  
E-mail: [water@mdpi.com](mailto:water@mdpi.com)  
[www.mdpi.com/journal/water](http://www.mdpi.com/journal/water)





MDPI  
St. Alban-Anlage 66  
4052 Basel  
Switzerland

Tel: +41 61 683 77 34  
Fax: +41 61 302 89 18  
[www.mdpi.com](http://www.mdpi.com)



ISBN 978-3-03897-549-6