A Machine learning approach (ANNs) for flood forecasting in Andernach station of River Rhine

Introduction

Floods are the most destructive natural disasters, causing chaos not only the human life but also the infrastructure, transportation networks ,crop loss and overall the socioeconomic system.[1] [2] . As a result of these concerns, experts are more interested in the prediction model's ability to reduce the aftereffects of incidences.[3]. It is necessary to analyze historical hydrological data to develop a prediction model that can be used to propose mitigation measures for a given area.

Machine learning (ML) methods have contributed significantly to the development of prediction models in recent years because they can easily simulate the complex mathematical expressions of physical flood processes, resulting in cost-effective solutions and improved performance. Furthermore, researchers are attempting to develop new machine learning methods as well as hybridize existing ones. Artificial neural networks (ANNs), neuro-fuzzy, support vector machine (SVM), and support vector regression (SVR) are just a several of the machine learning algorithms that have been reported to be effective for both short-term and long-term flood forecasting. [1]. Models based on artificial neural networks(ANNs) offer a significant opportunity to improve the efficiency of flood simulations and these models have a lot of potential for real-world applications like ensemble forecasting and uncertainty analysis.[4]

Objective

The main objectives of the research work are given below

- Timeseries analysis of rainfall-runoff parameters
- Develop an ANNs based machine learning model for predicting the discharge of River Rhine
- Performance analysis of the ANNs model with other Conventional models (Linear Regression, Decision Tree, Random Forest)
- Comparing simulated and observed data

Study Area

The Rhine is one of Germany's major rivers. Flooding is a common occurrence along the river's bank. The station of our study area Andernach is a 30,000-person town in the Mayen-Koblenz district of Rhineland-Palatinate, Germany. It is located on the left bank of the Rhine and elevation is 60m from the sea level. Because the Rhine widens near Andernach, this area was also flooded not only during the Eifel flood disaster of 2021 but also in previous years.

A Machine learning approach (ANNs) for flood forecasting in Andernach station of River Rhine

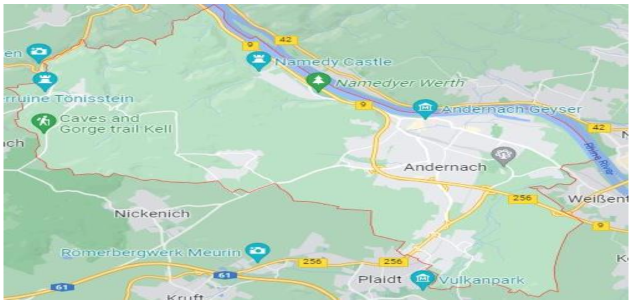


Figure 1: Andernach City,Germany

Research methods

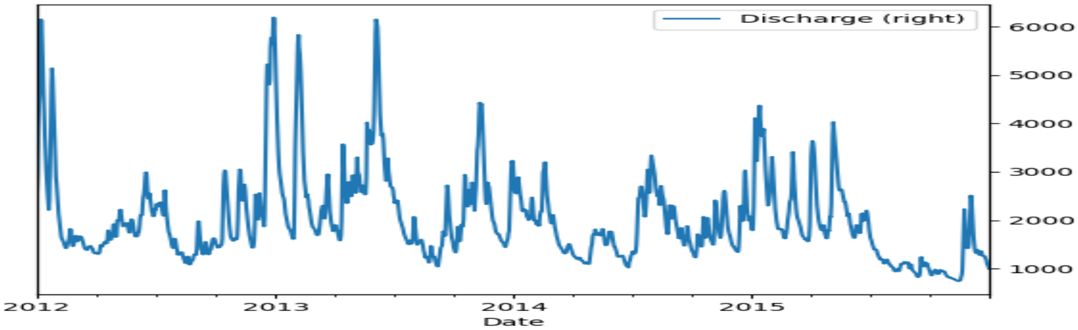
1.1: Preprocessing the Data

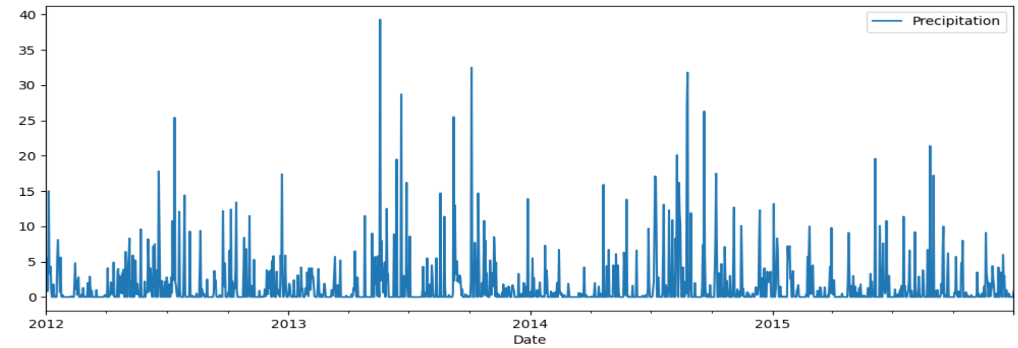
The accessible rainfall-runoff time series dataset (Precipitation, Evaporation, Temperature, Discharge) for Andernach Station from 2000 to 2018 are gathered from several websites, including https://www.wetter.rlp.de/, https://opendata.dwd.de/, and https://www.bafg.de/GRDC/.

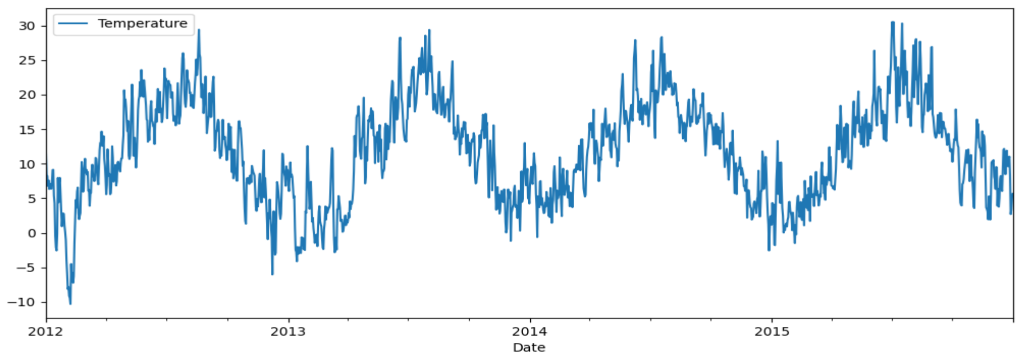
The raw data was cleaned in the initial stage. Outliers and gaps in the raw data existed; the gaps were filled by using Excel's built-in facilities and observing the patterns of neighboring data. then processing was done on the noisy data.

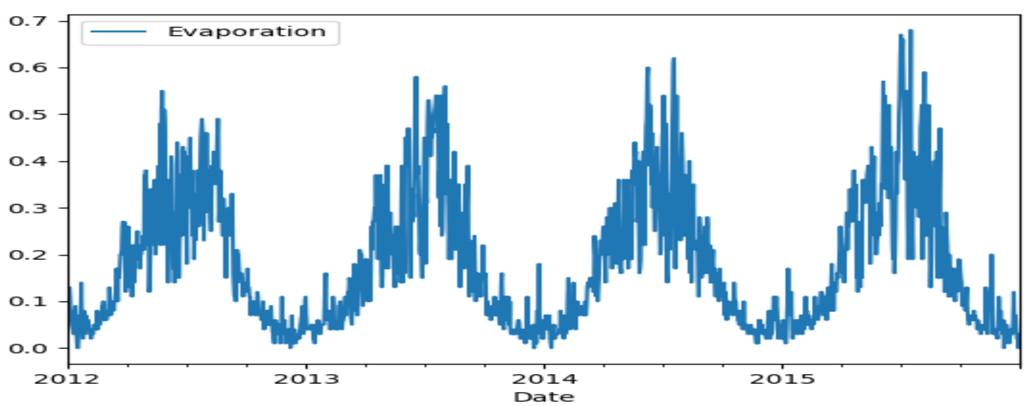
In the following step hourly data was converted to daily data for all datasets parameters and then 2012 to 2017 datasets are selected as processed data for the further analysis. In the next phase, all parameters of the dataset were changed from hourly to daily data, and then the year-specific datasets from 2012 to 2017 were chosen as processed data for the following study.

1.2 Visualization of datasets









1.3 Developing Artificial Neural Networks

Feed Forward Neural Network was decided to The Artificial neural networks structure of this study is done by python 3.10. At first the necessary libraries were installed such as keras, TensorFlow etc.

Although normalization is not required, it is recommended for neural networks since some activation functions are sensitive to the magnitude of numbers. The range of the standardization is 0 to 1 , and these should be done before being fed into a neural network [5].In the following step the whole dataset is split into two format ,one in training (about 66%of whole data) and another one is testing (about 34%of whole data).Below table is depict the dataset conditions in neural network environments.

Division	Percentage of the dataset	Timeseries (Approximate)	Total no of record
Training	66%	2012-2015	1446
Testing	34%	2016-2018	746
Total	100%	2012-2018	2192

Our data must be transformed into something resembling X and Y values. In this manner, a sequence rather than a collection of data points can be used to train it. divide the sequence of numbers divide into n columns for column X, where we will input the numbers, and use column Y as the final column, where we will output the subsequent number in the sequence. For this goal, a dataset matrix is created from an array of values. The seq size parameter determines how many previous time steps should be used as input variables to forecast the current time period. This parameter generates a dataset with two columns: Y, the number of discharges at time t+1, and X, the number of discharges at time t, t-1, t-2, etc. Below figure demonstrates the idea of the sequence

testX

	0	1	2	3	4
0	0.10341	0.11258	0.11441	0.10524	0.09608
1	0.11258	0.11441	0.10524	0.09608	0.09608
2	0.11441	0.10524	0.09608	0.09608	0.09608
3	0.10524	0.09608	0.09608	0.09608	0.09608
4	0.09608	0.09608	0.09608	0.09608	0.08874
5	0.09608	0.09608	0.09608	0.08874	0.08508

The raw data that is fed into the network is represented by the activity of the input units. The activities of the input units and the weights on the connections between the input units and the hidden units influence each hidden unit's activity. The activity of the hidden units and the weights between the hidden units and output units determine how the output units behave.

Figure 1 is the deception of most common type of artificial neural network, where it is observed that a network is made with number of nodes which is called neurons. These neurons are divided into three layers. The first one is input layers which is connected to hidden layers, and hidden layers is connected to the output layer. The input nodes work is to distribution of raw data to the network. the information passes from the input layer through the hidden layer and lastly to the output layer. This the basic network design of feed forward neural networks. And there is no regulation about how many nodes should be in a hidden layer. The activity of each hidden unit is influenced by the input units' activities as well as the weights on the connections between the input units and hidden units.

A Machine learning approach (ANNs) for flood forecasting in Andernach station of River Rhine

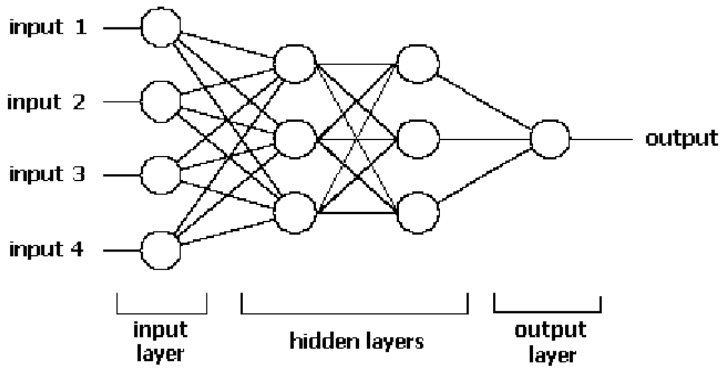


Figure: Simple Feed Forward neural networks [6]

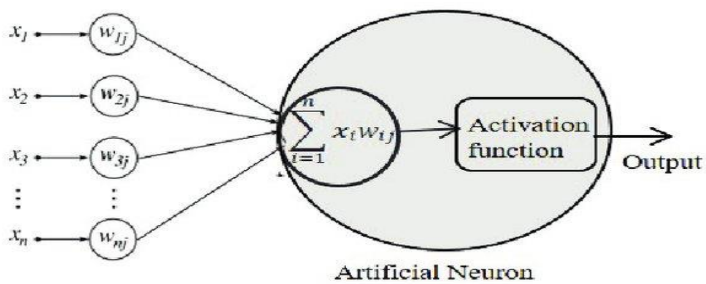


Figure: An Artificial Neuron[7]

Above figure describes a closer look of an artificial neuron, where each neuron j has inputs X_1, X_2, to X_n. And the weights of the inputs are W_{1j}, W_{2j} to W_{nj} . In a neuron j, the sum of all input is according to below equation. Here W_{oj} is an additional term called bias.

$$P_J = \sum_{i=1}^{n} X_i W_{ij} + W_{oj}$$

In order to get the final output from the neuron, an activation function is applied to the value P_J.The linear function is used as an activation function.

1.4: Performance Analysis of Model

As the network had been trained using calibration data, he result performance is evaluated by comparing its output with the validation set. The performance of the Model evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error(MSE) and a correlation coefficient in a comparative study (R) . Below equations represent the model performance analysis scenarios.

$$\mathrm{RMSE} = \sqrt{\frac{\sum(\mathrm{y_i} - \mathrm{y_p})^2}{\mathrm{n}}}$$

$$R^2 = \frac{\sum_{i=1}^{n}(\mathrm{y_i} - \mathrm{y_p})^2}{\sum_{i=1}^{n}(\bar{y} - \mathrm{y_p})^2}$$

$$\mathrm{MSE} = \frac{\sum(\mathrm{y_i} - \mathrm{y_p})^2}{\mathrm{n}}$$

$$\mathrm{MAE} = \frac{|\mathrm{y_i} - \mathrm{y_p}|}{\mathrm{n}}$$

$$\mathrm{y_i} = \text{Actual Value}$$

$$\mathrm{y_p} = \mathit{Predicted\ value}$$

$$\mathrm{n} = \mathit{number\ of\ observation}$$

$$\bar{y} = \mathit{Average\ value}$$

1.5: Comparing with conventional machine learning model

In this stage neural network outputs performance is evaluated with comparing the other conventional model

1.5.1: Linear regression

1.5.2: Decision Tree

1.5.3: Random Forest

A Machine learning approach (ANNs) for flood forecasting in Andernach station of River Rhine

Results:

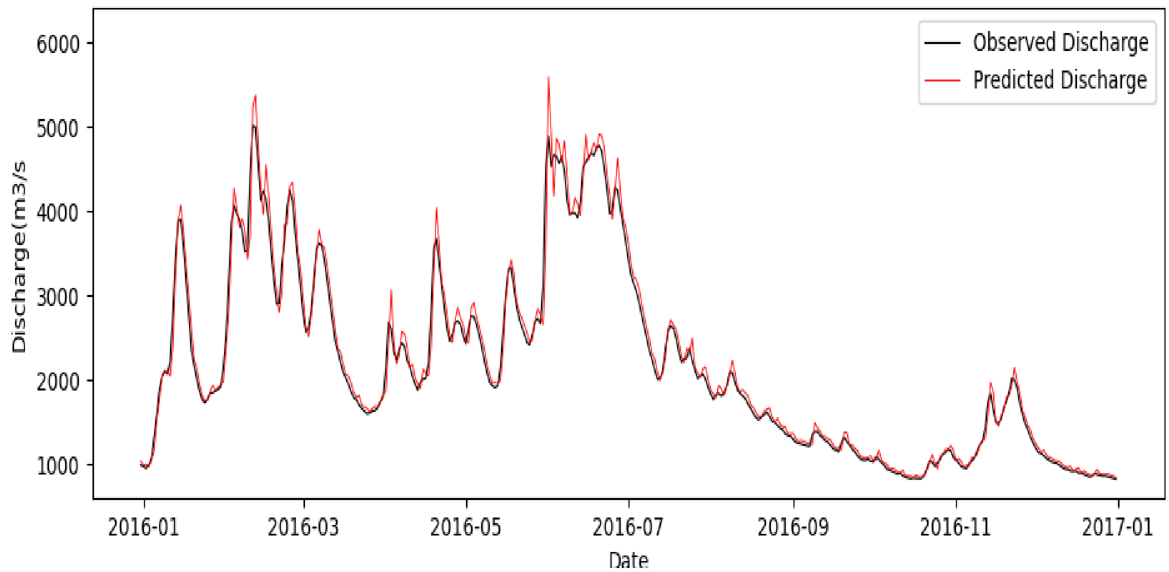


Figure: Predicted discharge (Test) for the year 2016

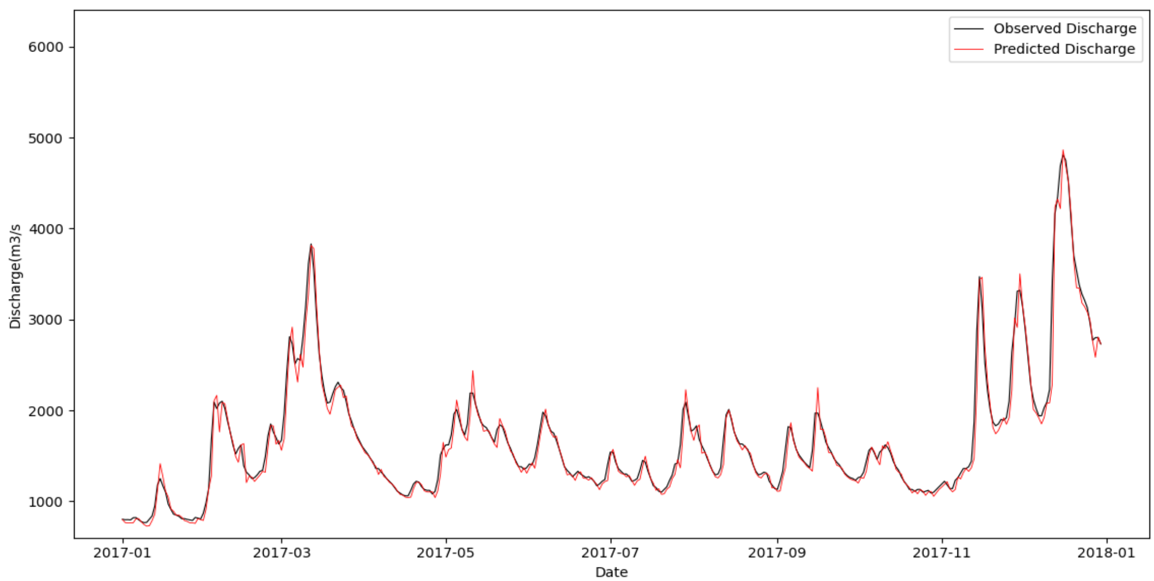


Figure: Predicted discharge (Test) for the year 2017

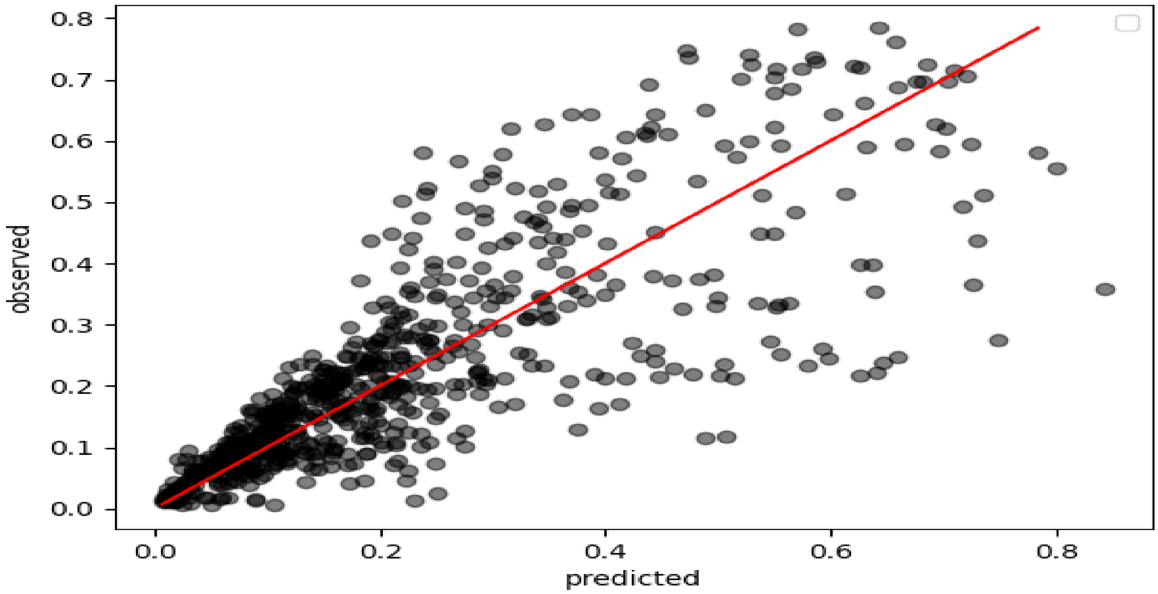


Figure 3: Scatter plot for observed vs predicted data

	Training/Calibration				Testing/Validation			
	R	RMSE	MAE	MSE	R	RMSE	MAE	MSE
ANN (Feed Forward)	.9816	121.55	0.0013	0.0004	.9799	129.71	0.0	0.0005
Linear Regression	.9824	.02	0.0130	0.0004	.9827	0.02	.0135	.0005
Decision Tree	1	0.00	0.00	0.0	0.9482	0.04	0.0222	0.0015
Random Forest	.9963	.01	0.0060	.0001	0.9690	.03	0.0175	.0009

Table: Model performance Analysis with other conventional model performance

A Machine learning approach (ANNs) for flood forecasting in Andernach station of River Rhine

Bibliography

1. Mosavi A, Ozturk P, Chau KW. Flood prediction using machine learning models: Literature review. *Water (Switzerland)*. 2018;10(11):1-40. doi:10.3390/w10111536
2. Elsafi SH. Artificial Neural Networks (ANNs) for flood forecasting at Dongola Station in the River Nile, Sudan. *Alexandria Eng J*. 2014;53(3):655-662. doi:10.1016/j.aej.2014.06.010
3. Obasi AA, Ogbu KN, Orakwe LC, Ahaneku IE. Rainfall-river discharge modelling for flood forecasting using Artificial Neural Network (ANN). *J Water L Dev*. 2020;44:98-105. doi:10.24425/jwld.2019.127050
4. Chu H, Wu W, Wang QJ, Nathan R, Wei J. An ANN-based emulation modelling framework for flood inundation modelling: Application, challenges and future directions. *Environ Model Softw*. 2020;124. doi:10.1016/j.envsoft.2019.104587
5. Dawson CW, Wilby R. *An Artificial Neural Network Approach to Rainfall-Runoff Modelling*.; 1998.
6. Augusto F, Fiorelli S, Tomita E, Neto AH, Campoleoni ET, Fiorelli FAS. Artificial Neural Network for Predicting Energy Consumption. doi:10.13140/RG.2.1.2758.0320
7. Kotteda VMK, Stephens JA, Spotz W, Kumar V, Kommu A. Uncertainty quantification of fluidized beds using a data-driven framework. *Powder Technol*. 2019;354(June):709-718. doi:10.1016/j.powtec.2019.06.021