**Developing a machine learning model (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.

![Map

Description automatically generated]()

Figure : Andernach City,Germany

**Research methods**

*Data availability*

The available time series of rainfall-runoff data set (Precipitation, Evaporation, Temperature, Discharge) of Andernach Station from 2000-2018 are collected from different websites like <https://www.wetter.rlp.de/> , <https://opendata.dwd.de/> , <https://www.bafg.de/GRDC/> .

*Analytical methods*

Basic target is set up an Artificial Neural Network model and comparing this model with another model like decision tree, linear regression, random forest etc. The performance of the Model will be evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and a correlation coefficient in a comparative study (R). After analyzing the available data set timeseries, 4-6 years data set will be selected for training another 1-2 years’ will be selected for testing the model.

**Figure**

**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**

**1.2 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

Table

Description automatically generated

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.

**Diagram

Description automatically generated**

Figure1:Simple Feed Forward neural networks 6

Diagram

Description automatically generated

Figure: An Artificial Neuron7

Above figure describes a closer look of an artificial neuron, where each neuron j have

igure 2 provides a closer look at an individual neuron (in the hidden and output layers). Each neuron, j , has a number of input arcs, w, to «„. Associated with each arc, i, is a weight, wip which represents a factor by which any values passing into the neuron are multiplied. A neuron, j , sums the values of all inputs according to equation (1): fi SJ='Zwijuj+w0J (1) In equation (1) an additional term, wop has been included called a bias. An activation function is applied to the value Sj, to provide the final output from the neuron. This activatio

#**Import the dataset and library**

**#Normalize the dataset**

**#Delimit Training and Testing**

**#Creating Sequences for the input of neural network**

**#Creating Architecture of Neural network**

**1.7: Performance Analysis of Model**

In order to train and test artificial neural networks it is necessary to have two sets of training data—a calibration set and a validation set. Having trained a network with calibration data the accuracy of the results obtained from that network can be assessed by comparing its responses with the validation set. In this study the comparison was made using the mean squared relative error (MSRE) calculated as the mean of the square of the errors relative to each actual expected value in the validation set and the root mean squared error (RMSE). Although the MSRE provides more meaningful measures of overall network performance than relative differences alone, RMSEs were also calculated to enable comparisons to be made with other model results cited in the literature. In addition, visual comparisons can be made by plotting observed and modelled results. Graphs showing the results of the ANN rainfall-runoff models are presented later.

**1.8: Plotting**

**1.9: Linear regression**

**1.10: Decision Tree**

**1.11: Random Forest**

**Results:**

**Equations**

**Results:**

Chart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, scatter chart

Description automatically generated

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Training/Calibration | | | | Testing/Validation | | | |
|  | R2 | RMSE | MAE | MSE | R2 | 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 |

After adding other features

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Training/Calibration | | | | Testing/Validation | | | |
|  | R2 | RMSE | MAE |  | R2 | RMSE | MAE |  |
| Linear Regression | 0.060 | 886.60 | 649.79 | 0.010 | 920.39 | 729.39 |
| Decision Tree | 0.993 | 71.83 | 7.67 | -0.400 | 1095.24 | 668.34 |
| Random Forest | 0.829 | 377.72 | 261.85 | 0.023 | 914.49 | 657.13 |

**Timeline**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | May | | June 2022 | | July  2022 | | | August  2022 | | September  2022 | | October  2022 | |
| Literature review |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data collection |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prepossessing and Data analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Develop the ANN model |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Develop the other ML models |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Comparing the models |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Report Writing |  |  |  |  |  |  |  |  |  |  |  |  |  |

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