

Article

Application of Machine Learning Techniques in Rainfall–Runoff Modelling of the Soan River Basin, Pakistan

Muhammad Tariq Khan ¹, Muhammad Shoaib ^{1,*} , Muhammad Hammad ¹, Hamza Salahudin ¹ , Fiaz Ahmad ¹ and Shakil Ahmad ²

- ¹ Department of Agricultural Engineering, Faculty of Agricultural Sciences & Technology, Bahauddin Zakariya University, Multan 60000, Pakistan; mtariqwws@googlemail.com (M.T.K.); hammadpattal93@gmail.com (M.H.); hamzasalahudin1@gmail.com (H.S.); fiazahmad@bzu.edu.pk (F.A.)
- ² NUST Institute of Civil Engineering, School of Civil & Environmental Engineering, National University of Sciences & Technology, Islamabad 44000, Pakistan; shakilahmad@nice.nust.edu.pk
- * Correspondence: msho127@aucklanduni.ac.nz or muhammadshoaib@bzu.edu.pk

Abstract: Rainfall–runoff modelling has been at the essence of research in hydrology for a long time. Every modern technique found its way to uncover the dynamics of rainfall–runoff relation for different basins of the world. Different techniques of machine learning have been extensively applied to understand this hydrological phenomenon. However, the literature is still scarce in cases of extensive research work on the comparison of streamline machine learning (ML) techniques and impact of wavelet pre-processing on their performance. Therefore, this study compares the performance of single decision tree (SDT), tree boost (TB), decision tree forest (DTF), multilayer perceptron (MLP), and gene expression programming (GEP) in rainfall–runoff modelling of the Soan River basin, Pakistan. Additionally, the impact of wavelet pre-processing through maximal overlap discrete wavelet transformation (MODWT) on the model performance has been assessed. Through a comprehensive comparative analysis of 110 model settings, we concluded that the MODWT-based DTF model has yielded higher Nash–Sutcliffe efficiency (*NSE*) of 0.90 at lag order (Lo4). The coefficient of determination for the model was also highest among all the models while least root mean square error (*RMSE*) value of 23.79 m³/s was also produced by MODWT-DTF at Lo4. The study also draws inter-technique comparison of the model performance as well as intra-technique differentiation of modelling accuracy.

Keywords: machine learning; ANN; single tree boost; decision tree; decision tree forest; Pothohar region



Citation: Khan, M.T.; Shoaib, M.; Hammad, M.; Salahudin, H.; Ahmad, F.; Ahmad, S. Application of Machine Learning Techniques in Rainfall–Runoff Modelling of the Soan River Basin, Pakistan. *Water* **2021**, *13*, 3528. <https://doi.org/10.3390/w13243528>

Academic Editor: Kwok-wing Chau

Received: 20 October 2021

Accepted: 3 December 2021

Published: 9 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The predominant role of several nonstationary and nonlinear variables in transformation of rainfall into runoff makes it difficult to comprehend [1]. Moreover, the spatiotemporal variability in rainfall intensity and uniformity adds to the complexity of modelling basin response to catchment precipitation [2]. However, the direct involvement of rainfall in runoff generation and runoff in streams, rivers, and even floods, makes it one of the most focused hydrological phenomena. The natural disasters such as fluvial and pluvial floods and hydrological droughts are also determined by the rainfall–runoff relationship of any basin [3]. Therefore, precise and accurate rainfall–runoff modelling is important for effective management of water resources and prediction and prevention of natural calamities. The relationship between rainfall and runoff is also empirically expressed in the mass balance equation [4]. From the equation it can be observed that both rainfall/precipitation (*P*) and runoff (*R*) play a decisive role in storage change (*dS*), while infiltration (*I*) and evaporation losses (*E*) are also to be considered for an accurate estimation of change in water storage.

$$dS = P - R - I - E \quad (1)$$

Owing to the direct participation of rainfall in the generation of runoff, hydrologists and researchers worldwide have been trying to understand the exact relationship between the two hydrological variables. Rainfall–runoff modelling has been the prime focus of hydrology research for a long time. It has brought the research area to an advanced phase of maturity and diversity in context of mathematical and methodological background [5]. However, the remaining discrepancy in rainfall–runoff relation and the emergence of machine learning is promising. These computational techniques either reduce the parametric requirement or enhance modelling accuracy, or even serve both purposes [6]. The rational method of runoff estimation by Mulvaney [7] is often rated as the first attempt of rainfall–runoff modelling [6]. However, significant advances in different conceptual and physical models were introduced in the field of hydrology. The mean annual method [8], unit hydrographs [9] and instantaneous unit hydrographs [10], Stanford Watershed Model (SWM) [11], and Huggins–Monke model [12] are some of the significant names.

The initial application of machine learning in rainfall–runoff modelling [13] and rainfall forecasting [14] set modelling in hydrology to a whole new direction. Artificial neural networks (ANN) gained massive popularity in rainfall–runoff modelling [2,15–22] and rainfall forecasting [23–29]. Other techniques of machine learning, such as gene expression programming (GEP), single decision tree (SDT), decision tree forest (DTF) and tree boost (TB), were also extensively used by researchers [30–32].

Earlier studies compared machine learning techniques with conventional conceptual or physical-based models while the recent studies probed deeper into machine learning models for better results. For example, Shoaib et. al [21] compared the performance of wavelet-based multilayer perceptron neural networks (MLPNN) and radial basis function neural networks (RBFNN) in rainfall–runoff modelling of the Brosna River basin. The authors suggested that data pre-processing with wavelet transformation significantly enhanced modelling efficiency of both techniques. Similarly, Asadi et. al [15] pre-processed ANN inputs using a standardization process and enhanced the performance of simple ANN models in the Houghton and Calliope river basins. Kumar et. al [17] contrasted the performance of emotional neural networks (ENN) and typical ANN models in rainfall–runoff modelling. The authors concluded that ENN outperformed ANN rainfall–runoff modelling of Sone command, Bihar. Sharghi et. al [20] also implemented an emotional artificial neural network (EANN) and compared its performance with wavelet-based ANN (WANN) models. This study also marked EANN to be better than ANN and WANN in rainfall–runoff modelling application. In another study, Sharghi et al. [33] coupled EANN models with wavelet transformation (WEANN). The authors observed that WEANN outperformed WANN and EANN when applied to the rainfall–runoff modelling problem of the Trinity River basin. Similarly, Poonia and Tiwari [18] implemented newer variants of ANN, RBFNN, and feed-forward back propagation model (FFBP) for modelling the Hoshangabad Basin, Narmada River. The authors observed that RBF yielded higher modelling as compared to FFBP models.

Contrastingly, other data-driven soft computing techniques, including DTF, TB, SDT, and GEP, have confined application in rainfall–runoff relation extraction problems. However, these techniques have been used for different hydrological problems. Aytok et al. [34] employed GEP models for rainfall–runoff simulation of Juniata River basin in the USA. The authors contrasted model performance with that of ANN variants, FFBP, and generalized regression neural networks (GRNN). It was concluded that GEP performed well in comparison with ANN models, and thus could be used as an ANN alternative subject to fulfilment of parametric requirement. Nourani et. al [35] coupled M5 decision tree with wavelet transformation for rainfall–runoff modelling and compared its performance with simple ANN and M5 tree models. In conclusion, the wavelet coupled M5 decision tree model performed better than both the ANN and M5 tree models. Lee et al. [36] applied DTF and TB techniques in spatial mapping of flood susceptibility in Seoul, South Korea. It was observed that the machine learning techniques performed better than other conventional methods used in the study. Similarly, Raza et al. [37] used SDT, TB, and DTF, to estimate

reference evapotranspiration in different climate zones of Pakistan. TB models yielded better results than SDT and DTF in most locations, while in general, machine learning techniques outperformed conventional methods of evapotranspiration estimation, such as the Penman–Monteith (FAO-56 method) [38].

A comprehensive literature review of above-cited streamline studies exhibits a clear gap in terms of an encompassing research work that compares the performance of major machine learning techniques in rainfall–runoff modelling. Moreover, the effect of data pre-processing with wavelet transformation on the performance of GEP, TB, SDT, and DTF is still unclear. To address this research gap, the main objectives of this study are: (1) to apply the major machine learning techniques ANN, GEP, SDT, TB, and DTF in a single research work, (2) to couple every technique with wavelet transformation in order to assess the effect of pre-processing on model performance. This study provides comprehensive research on machine learning application in rainfall–runoff modelling along with implementation of wavelet pre-processing. Other machine learning approaches include regression models (linear and logistic), support vector machine (SVM), k-nearest neighbour (KNN), naive Bayes, and gradient boosting models, but the selection of abovementioned techniques (ANN, GEP, SDT, TB, and DTF) for analysis is due to their up-to-date modelling variants, and overwhelming and continuous application in hydrological and environmental problems. The selected techniques have been separately studied in hydrological context in earlier studies of various authors [21,22,25,39,40]. However, basin-wide study encompassing ML techniques and wavelet transformation in the Pothohar region is a novel attempt provided in this research report.

2. Methodology

2.1. Study Area and Datasets

Largest of six basins in Pothohar region, the Soan River basin (Figure 1) spans over 9994 km² in Pakistan's sub-Himalayan Pothohar region with elevations ranging from 222 to 2261 m (a.s.l.) [41]. Islamabad, Attock, Chakwal, and Rawalpindi subdivisions have administration authority over the region. The region has continental and subtropical climate with scorching summer and relatively cold winters with semiarid climatic conditions. The coldest month is December, with an average temperature of 9 °C, while the hottest month is June, with an average temperature of 31 °C. The average yearly rainfall in the region varies from 400 mm to more than 1710 mm in hilly areas, with about 65% of it occurring within the monsoon season. Precipitation and permanent streams held in very small dams are essential for agricultural practices in the region. Fodder, wheat, oil seeds, chickpea, millet, peanut, and sorghum are the main crops cultivated in rainfed conditions. According to land use research, farmland accounts for 10% of the total area, while composite forestry accounts for 8% of the total land area. The most acreage is occupied by pasture and barren ground (49%), followed by hilly area featuring brush and shrubs (30%). Domestic areas account for 2% of the total area, whereas water bodies account for approximately 1% [42]. Soil formation is typically noncalcareous, thick, and diverse in type, ranging from clay loam to silty clay loam with adequate permeability, and are found on floodplain and loamy lands. According to the slope categorization, 52% of the region is level to mild (5° slope), whereas 22% is moderate (5–15° slope). Steep (15–30°) and extremely steep (greater than 30°) gradients account for 19% and 7% of total area, correspondingly [43,44]. Generally, moist and subhumid climates predominate in the northern portion of the region, whereas dry and semiarid environments predominate in the centre and southern regions, correspondingly. The basin is bounded by the Murree and Margalla Hills on north side, while the southern perimeter is bounded by the Salt Range. The other five river basins (Haro, Bunha, Kahan, Kashni, and Reshi) of the Pothohar region are also geographically and environmentally significant for the climatic and hydrologic dynamics of the region. The selection of Soan River basin over other basins is due to the large area of the basin and highest number of rain gauge stations. For better understanding of hydrology of the

region, all the six basins should be studied separately instead of one generalized study of whole region. Figure 2 presents the elevation model of the study area.

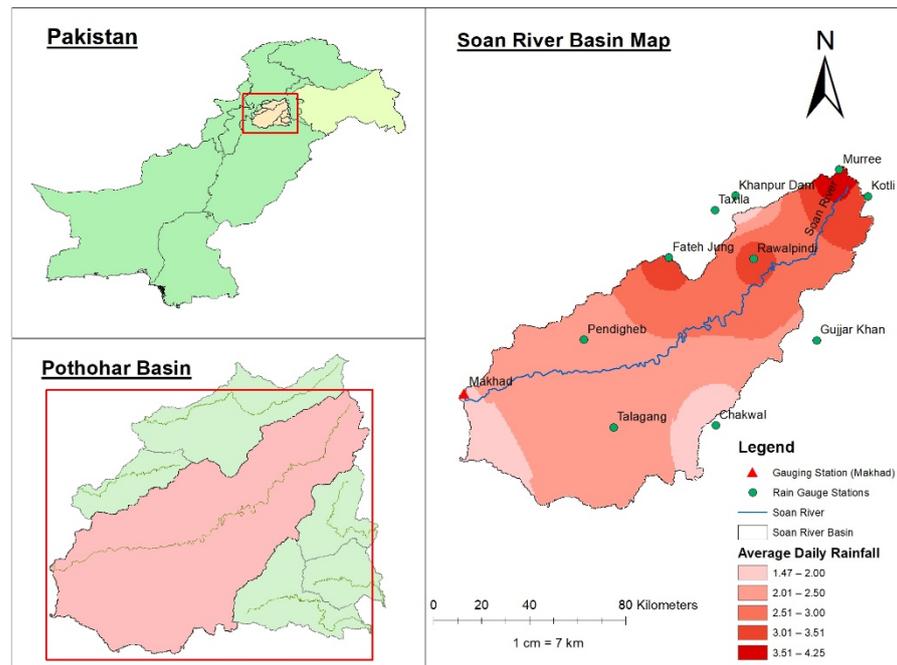


Figure 1. Map of the Soan River basin and location of rain gauges and gauging station.

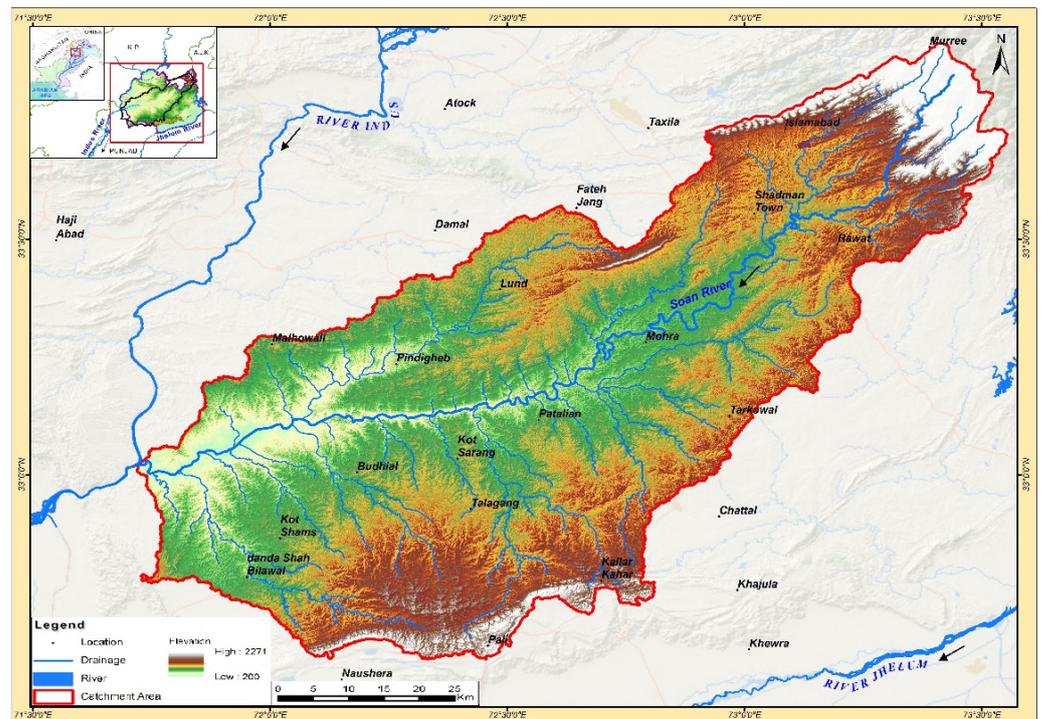


Figure 2. Digital elevation model (DEM) of the Soan River basin.

Data Acquisition

Daily rainfall and runoff data of 19 years (1999–2017) were acquired from three agencies (Pakistan Meteorological Department (PMD), Water and Power Development Authority (WAPDA), and Soil and Water Conservation Research Institute (SAWCRI)) in Pakistan. Rainfall data included the daily recorded data from 10 stations in the basin,

which were then aggregated to calculate the daily basin rainfall that was used as model input. The rainfall data from rain gauge stations was preferred over the satellite-derived rainfall data due to the ease of obtaining data and its reliability with lesser chances of errors within it. Likewise, daily discharge data (m^3/s) recorded at Makhad, Pirpiyahi (gauging station of Soan River basin), was acquired from WAPDA. Recent observations (2017–2021) were excluded from the analysis due to various inconsistencies in data at multiple stations and unavailability at a very few stations. Therefore, including incomplete data would have affected the reliability of outcomes. However, no significant change in data trend was observed from 2017 to 2021. Statistical characteristics of the datasets are summarized in Table 1, while Table 2 presents the summary of rain gauge and gauging station attributes.

Table 1. Statistical characteristics of datasets.

Datasets	Min.	Max.	Mean	Mode	S.D.
Training (1999–2012)					
Rainfall (mm)	0	95.48	2.61	0	5.78
Runoff (m^3/s)	2.16	1407.69	25.49	4.90	74.00
Testing (2013–2017)					
Rainfall (mm)	0	74.60	3.08	0	6.03
Runoff (m^3/s)	1.96	1559.74	33.42	5.89	72.81

Table 2. Attributes of rain gauge and discharge stations.

Serial No.	Station Name	Latitude ($^{\circ}\text{N}$)	Longitude ($^{\circ}\text{E}$)	Altitude (m)	Agency
1	Murree	33.9070	73.3943	2025	PMD
2	Rawalpindi	33.5651	73.0169	540	PMD
3	Kotli Sattian	33.8082	73.5255	1352	PMD
4	Chakwal	32.9328	72.8630	522	PMD
5	Fateh Jung	33.5635	72.6375	514	SAWCRI
6	Talagang	32.9172	72.4081	457	SAWCRI
7	Gujjar Khan	33.2616	73.3058	458	WAPDA
8	Pendigheb	33.2452	72.266	310	WAPDA
9	Taxila	33.7463	72.8397	549	WAPDA
10	Khanpur Dam	33.8018	72.9305	545	WAPDA
11	Makhad (gauging station)	33.0281	71.7393	252	WAPDA

The data were divided into two subsets, i.e., 70% of the data, having 5114 observations, was used for training of models while the remaining 30%, having 1826 observations, was used for validation, and testing of the developed models with each of the two sets having 15% of the total observations. The training models were first validated against validation set and then tested against the tested set. This approach limited the chances of an overfitting problem. This data partition is a simple iteration of the V-fold data division rule [45] and is generally practiced in this area of research [46–48].

2.2. Single Decision Tree (SDT)

The SDT is a method that displays a dataset's treelike structure, shown in Figure 3. Because it depicts a multidirectional splitting mechanism, SDT is also regarded as a logical data model. The SDT's data flow structure consists of rectangular boxes called nodes. Every node represents the numbers of rows present in the provided collection of data. Rainfall is selected as the predictor variable while runoff is designated as the target variable. In the present study, rainfall is also used as the root node because only one predictor variable is used. Additionally, entropy (E) and information gain (IG) of rainfall data are calculated to validate root node selection. After that, nonterminal and terminal nodes are created from the root node. Next, child nodes are produced by taking the decision of the terminal

nodes. SDT models are thus finalized to predict the target variable. End block represents the finalization of model construction when information gain reaches a maximum set level. The output from the model is simulated runoff in response to the predictor variable (rainfall), which is then compared with observed runoff for performance evaluation.

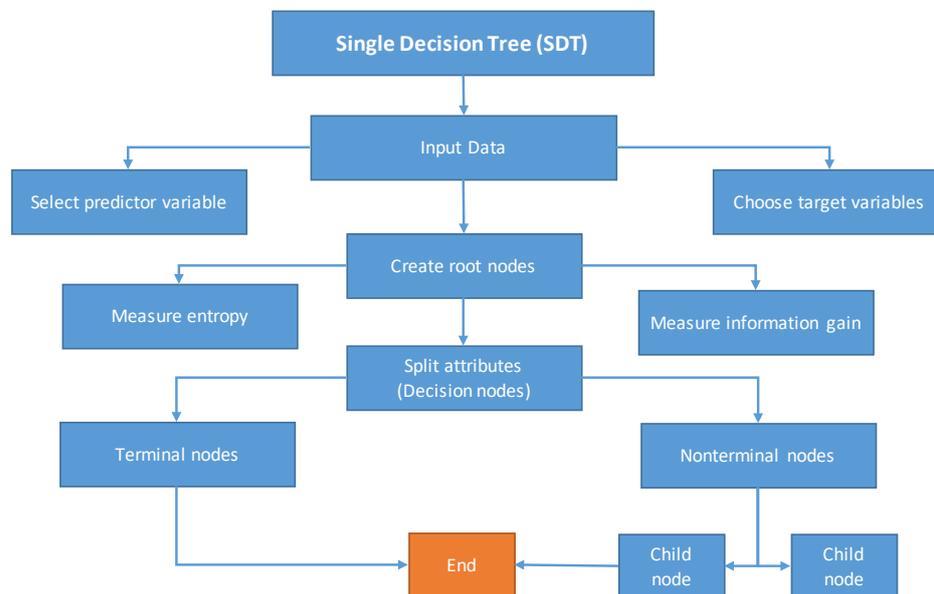


Figure 3. Single decision tree (SDT) flow chart (reproduced from Raza et al. [37]).

The ID3 algorithm considers “entropy” (E) and “information gain” (IG) estimation to be a critical and necessary phase. Entropy (E) may be defined as “evaluating training set impurity” or “assessing the uniformity of a training set,” whereas information gain (IG) can be defined as “assessing the anticipated decrease in entropy” or “evaluating the clarity of a training set.” The following equations [37] show the generic empirical formula of E and IG :

$$E = -A \log_2 A - B \log_2 B \tag{2}$$

$$IG = E_b - E_a \tag{3}$$

where E is the entropy of data and IG is the information gain of the variable. E_b is the entropy of distribution before the split and E_a is the entropy of distribution after the split, whereas A and B are the values of two choices for which the decision is made. According to Kotsiantis [49], SDT is a supervised learning technique that may potentially be used to solve classification and regression tasks.

2.3. Tree Boost (TB)

In the tree boost technique formulated by Breiman [50], a prediction algorithm is enhanced by using it on multiple repetitions in the boosting method. Afterward, every function’s outcome is added together to determine its weight. This step yields greater accuracy in predictions and removes the inaccuracy that occurs throughout the procedure. The separation of independent and dependent components is a better characteristic of this approach. The TB framework is mathematically represented in Equation (4), which is written as [37]:

$$\text{Predicted value} = R_0 + C_1 \times t_1 (X) + C_2 \times t_2 (X) + \dots + C_n \times t_n (X) \tag{4}$$

where R_0 denotes the initial value of the input series, C_n is the coefficient of tree nodes ranging from 1 to n , X is the vector of residuals, and $t_n (X)$ is the number of trees fitted to the residual vector.

The TB model, in the present study, was created using rainfall as the predictor variable and observed runoff as the target variable. In the boosting procedure, which involves connecting three sets of trees in sequence, a basic tree ensemble process is thus developed. Tree 2 receives the residuals produced by Tree 1. Similarly, Tree 3 includes Tree 2's residuals. Between Tree 1 and Tree 3, there exists no straight relationship. This procedure repeats until the whole dataset have actively taken part in the modelling. The number of trees in series is set to a range of 20 to 400, and the split size is set as 10 with a trimming factor of 0.01. The loss function (Huber M-regression) is used for disregarding the misclassification error. The threshold cut-off is set between 0.90–0.95, which means the trimming of series continues until the classification accuracy reaches the set value. After that, the model output is generated, and the output is compared with the target variable for performance evaluation. The selected parameters can also be iterated for training new models on different datasets. Change in split size, trimming factor, loss function, and threshold value can be used iteratively to formulate new models. The general architecture of a tree boost model is shown as a flow chart in Figure 4.

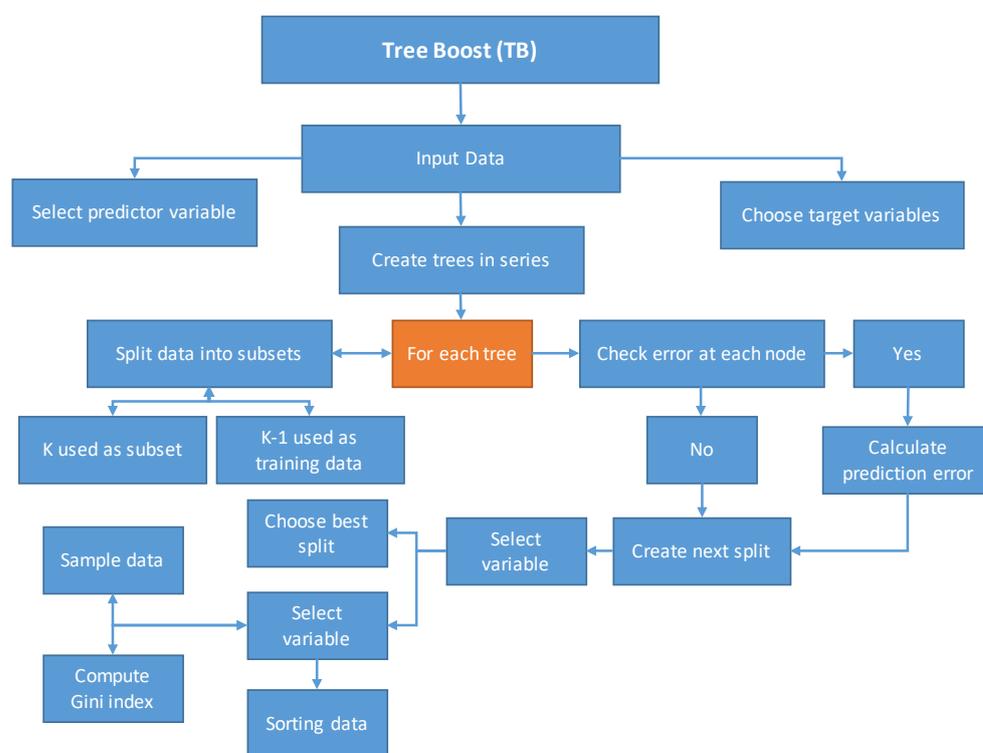


Figure 4. Tree boost (TB) flow chart (reproduced from Raza et al. [37]).

2.4. Decision Tree Forest (DTF)

The DTF is a type of machine learning method that relates to a certain category. It consists of a number of decision trees that aggregate the predicted values to provide an overarching prediction for the given dataset. In the DTF method, the numbers of trees develop in parallel to one another without any connection. The DTF technique is based on Breiman's [51] initial concept of the random forest (RF) technique.

First, rainfall is set as the predictor variable from which training set and out-of-bag (OOB) data sets are generated. Training data actively take part in the analysis while OOB data do not contribute in it and rather used as a validation set for the model. Observed runoff is set as the target variable. Assume that the entire set of data consists of N occurrences. Given the N occurrences, the desired fraction is selected from the given dataset. Bagging is the term for this procedure. Since chosen data are utilized multiple times in the bagging procedure, all of the information may not always engage in the testing. In general, 67% of the data set values are utilized as data sample, with the remaining 33%

being “out-of-bag” (OOB) data. The bagging procedure is repeated until all the trees are created. In the development of the trees, one sample of predictor variables is chosen from the entire collection of predictor variables. Model parameters involve trees in forest, split size of each node, and depth of individual trees, which are initially set by default and then changed by a trial-and-error method. To begin, each tree is given a number of rows within data set as input data, and then the predictor variable is selected to divide every node of the tree. When coupled with predicted values, these extra characteristics substantially enhance the aggregate prediction/forecasting accuracy of the DTF models. The split information of the predictor variable values indicates the impact of these values on the predicted value. This outcome is then compared with the target variable for performance analysis of the model. Figure 5 shows the working flow chart of a general SDT model.

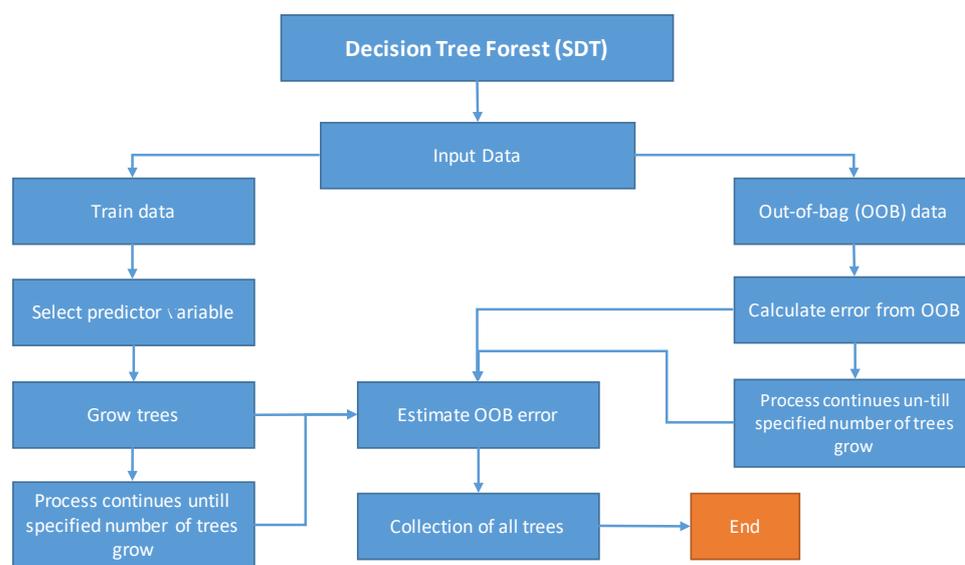


Figure 5. Flow chart of decision tree forest (DTF) (reproduced from Raza et al. [37]).

2.5. Multilayer Perceptron (MLP)

MLP, formulated by Webros [52], is a kind of fully connected feed-forward neural network (ANN) that is made up of layers of neurons. MLP comprises, at minimum, three levels: an input layer, one or more hidden layers, and an output layer. The output layer consists of one neuron, which represents the MLP ANN’s output—in this case, the estimated runoff in m^3/s . The number of neurons in the input layers is equal to that of inputs in the dataset [53].

MLP works by computing the value of neurons in an active layer through the activated aggregate of weighted output of neurons linked to the neuron in a preceding layer [54]. The values of weighted inputs are used as inputs to the neuron activation function, which either outlines the inputs to outputs instantly (i.e., identity activation function), inside specified boundaries (i.e., sigmoid, or tanh), or whilst also eliminating unnecessary attributes (i.e., ReLU). The weights of the neuron are typically arbitrary, but they are modified via a backward propagation of error from the output layer to the input layer, and then weights are modified proportionately to the error. The feed-forward component of the models allows information to move from input layer to the output layer in a forwarding manner, while the back-propagation method allows the error to propagate backwards, which changes and modifies the connection weights [25]. This combination of forward and backward flow creates a two-way passage for information moving forward and error moving backward, thus resulting in better outcomes.

An MLP structure with one hidden layer is shown in Figure 6; n denotes the number of neurons in the input layer. Similarly, h represents the number of hidden layer neurons while the output layer contains only one neuron that is denoted by n_0 . $R, R_{t-1}, R_{t-2}, \dots,$

R_{t-d} , are inputs to the model that resemble the case examined in this study; w_{13}^H symbolizes the connection weight of the first neuron in the hidden layer to the third neuron in the input layer. Likewise, w_{12}^O is the weight connecting the first neuron in the output layer to the second neuron in the hidden layer. All the pertinent connection weights are designated with a particular code, as shown in Figure 6. A generalized MLP model is numerically articulated by Babs [55] as:

$$Q_t = F_2 \left[\sum_{j=1}^h W_{kj} F_1 \left(\sum_{i=1}^n W_{ji} x_i + b_{j0} \right) + b_{k0} \right] \quad (5)$$

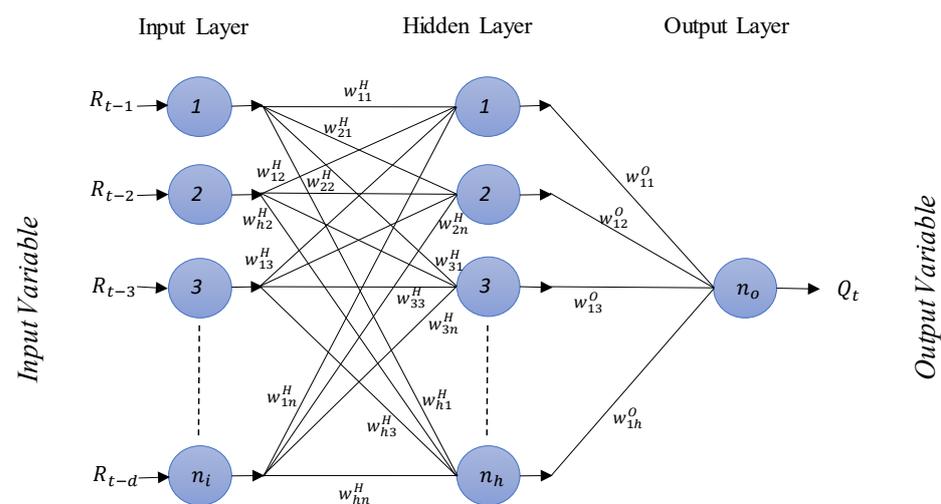


Figure 6. Structure of a general multilayer perceptron (MLP) ANN.

Here, Q_t is the output of the model that represents the predicted runoff at time t by using rainfall at time $t-d$ as input variables. R_{t-1} to R_{t-d} is input to the i -th neuron in the input layer, Q_t is the output from the k -th neuron in the output layer, F_1 is the hidden layer activation function and F_2 is the output layer linear activation function; b_{j0} and b_{k0} are the bias units of the j -th neuron in input layer and the k -th neuron in the output layer, respectively, and W_{ji} denotes the connection weight of the i -th neuron in the input layer and the j -th neuron in the output layer.

2.6. Gene Expression Programming (GEP)

GEP is a technique for creating computing programmes and quantitative models that are influenced by spontaneous evolution and relies on adaptive computing. It uses a dataset to provide a result in the shape of tree architecture. Ferreira developed this technique in 1999 and officially presented it in 2001 [56]. The GEP technique combines the main views of the two previous inheritance methods (“genetic algorithm” GA and “genetic programming” GP) in order to solve their flaws. The chromosomal genetic constitution in this technique is comparable to that of a GA, while the phenotype of a chromosome is a tree data structure with dynamic range and height, comparable to that of the GP technique. As a result of eliminating the previous methods’ double function constraint of a chromosome, the GEP algorithm allows many genetic algorithms to be guaranteed of the offspring’s chromosomes’ stable viability at a quicker pace than GP. If there are many parameters, the rational connection between them may constitute a relationship which can be properly stated.

Employing the GEP method, a collection containing linear chromosomes is initially generated to determine the connection between parameters a and b and y . Any of the parameters can be put at every location of the genes on the chromosomes. It is essential to evaluate the viability of chromosomes once they are produced and respective places occupied. The chromosomes are represented as “expression tree” (ET) in the GEP method.

2.7. Maximal Overlap Discrete Wavelet Transformation (MODWT)

Following the initial conceptual development, wavelet transformation has received a much recognition as a sophisticated type of mathematical signals analysis method [57]. Combining frequency spectrum analyses and temporal encoding of the data, wavelets are a useful approach for retrieving latent knowledge and characteristics in unprocessed data sets [22]. It has the potential to uncover underlying characteristics such as patterns, inconsistencies, extremes, and collapse points in nonstationary data sets. Any alternative technique will struggle to incorporate these intricate features of nonstationary data [46]. Therefore, wavelet pre-processing extracts additional information from a raw signal that is then used by the models to generate better results as compared to models with raw inputs. A wavelet is a short-duration sinusoidal fluctuation with a zero-mean magnitude, as defined above.

In this research, an improved version of WT called the “maximal overlap discrete wavelet transformation” (MODWT) was utilized to divide a time series or a specified variable into scaled parts. It has a major advantage over traditional DWT in that it eliminates down samples [58] and limits the amount of inaccuracies due to time series overtranslation by rectifying the boundary of the transformed data series [48,59]. As shown in Figure 4, an unprocessed input (x_n) is processed across two complementary filters, a low pass filter (\tilde{g}) and a high pass filter (\tilde{h}), before being decomposed into two signals: approximations (a) and details (d). The low pass filter disregards high frequency components of the time series and allows only high-scale low-frequency portions only (i.e., approximation components), which are considered to be more important as they represent the signal’s identity and shape [60]. Likewise, a high-pass filter accepts low-scale high-frequency components (i.e., detail components) that are the white noise and nuance present in the rainfall data. The original signal can be reconstructed by adding detailed components of each level and approximation component of the highest level.

Equations (6) and (7) may be used to determine the wavelet coefficient (\tilde{W}_j) and scaling coefficient (\tilde{V}) for the n -th component at the j -th level [48]:

$$\tilde{W}_{j,t} = \sum_{l=0}^{L-1} \tilde{h}_l \tilde{V}_{j-1, (t-2^{j-1}l) \bmod N} \tag{6}$$

$$\tilde{V}_{j,t} = \sum_{l=0}^{L-1} \tilde{g}_l \tilde{V}_{j-1, (t-2^{j-1}l) \bmod N} \tag{7}$$

where t denotes the time factor and $\tilde{V}_{0,t}$ represents the original signal. $\tilde{V}_{j,t}$ and $\tilde{W}_{j,t}$ are the scaling and wavelet components of MODWT, correspondingly, and similarly \tilde{h} and \tilde{g} , respectively, relate to the wavelet and scaling filters. L denotes the length of wavelet filter while N is the module operator. Figure 7 shows the general structure of the MODWT model.

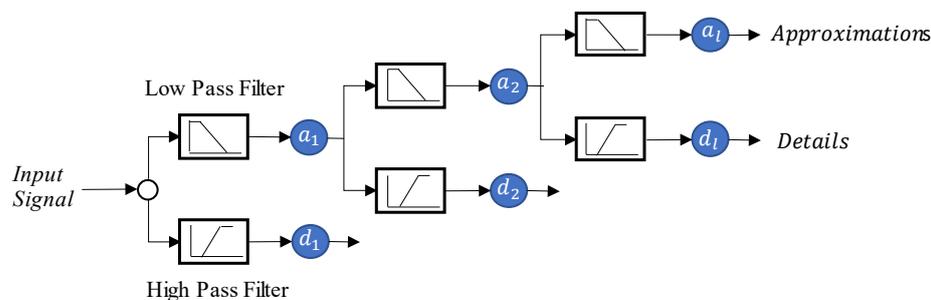


Figure 7. MODWT structure with low-pass and high-pass filters (reproduced from Hammad et al. [25]).

Other well-known alternative techniques of data/signal pre-processing include Fourier transform (FT) [61], bootstrapping, clustering, and classification models. The selection of

wavelet transformation (WT) for data pre-processing is due to the fact that WT is established to outperform previous methods including FT, as FT modifies data in frequency domain only while WT uses frequency and time domain analysis to extract more information from the raw signal.

2.8. Model Selection

In this study, five well-known and established machine learning techniques (SDT, TB, DTF, MLP, and GEP) were used to perform rainfall–runoff modelling of the Soan River basin. Daily rainfall and runoff data spanned over 19 years from 1999 to 2017. Data were divided into two subsets, namely, a training set and testing set. Seventy percent of the data were used for training and validation of models while 30% was used for testing of the models. The study also aimed to assess the effects of data pre-processing on all aforementioned techniques; therefore, two types of models were formed for each technique. Initially, rainfall at time “ t ” was used as input and runoff data were used as a training target. Initial models of each technique were trained and tested using original time series. Then, multiple models of each technique were generated by lagging the inputs up to lag order (Lo) 10. Table 3 summarizes the input models based on Lo. Here, Lo1 has input variable R_{t-1} that represents the rainfall lagged one day. Similarly, Lo2 having input variables R_{t-1} and R_{t-2} that represent the rainfall lagged up to one day and two days. Likewise, Lo10 is the rainfall time series lagged from day 01 to day 10. The introduction of time lag (in days) in models was done to study the temporal correlation of rainfall. Selection of optimum lag order range (from Lo0 to Lo10) was done by trial-and-error method and then further verified by the partial correlation method for input optimization [62]. Another technique for Lo optimization is particle swarm optimization emphasized by Ribeiro et al. [63].

Table 3. Input models based on time lag order (Lo).

Lag Order (Lo)	Input Variables
Lo0	R_t
Lo1	R_{t-1}
Lo2	R_{t-1}, R_{t-2}
Lo3	$R_{t-1}, R_{t-2}, R_{t-3}$
Lo4	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}$
Lo5	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, R_{t-5}$
Lo6	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, R_{t-5}, R_{t-6}$
Lo7	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, R_{t-5}, R_{t-6}, R_{t-7}$
Lo8	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, R_{t-5}, R_{t-6}, R_{t-7}, R_{t-8}$
Lo9	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, R_{t-5}, R_{t-6}, R_{t-7}, R_{t-8}, R_{t-9}$
Lo10	$R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, R_{t-5}, R_{t-6}, R_{t-7}, R_{t-8}, R_{t-9}, R_{t-10}$

The input variables of the models (Table 3) depend on the final lag order. Thus, in an input model with an n th lag order, there would be “ n ” input variables ($R_{t-1}, R_{t-2}, R_{t-3}, \dots, R_{t-n}$). Alternatively, input combination can also be formed in such a way that each input model contains a single variable consisting of the rainfall series lagged to the relevant Lo, for example, R_{t-1} at Lo1, R_{t-2} at Lo2, R_{t-3} at Lo3, and, similarly, R_{t-n} at Lo n . The selected combinations ensure that influence of previous events is transferred to the next lag while the alternative combinations are confined to single-day rainfall events only.

After simple models of each technique were trained and tested, all the inputs in Table 3 were pre-processed using MODWT. To select the optimum wavelet function and decomposition level, the initial MODWT-based model was transformed with all wavelet filters of Daubechies, finding literary support from earlier studies [21,64]. The db4 wavelet function outperformed other filters, which was then used to select the optimum level of decomposition using a trial-and-error approach. After comparative analysis of db4 at all decomposition levels, it was observed that db4 at decomposition level 3 was the optimum wavelet function. It was then used to transform generated inputs for each technique. Alternative wavelet families are Haar, Symlets, Sine, Spline, and Coiflet, but

Daubechies (db) is known to outperform other wavelet families in similar hydrologic problems. Therefore, the db family is selected while wavelet function (db4) is selected on performance basis. Table 4 presents the summary of all parameters, which were used during training and testing of all the techniques that were adopted in this study. Optimum parameters were selected with the combination of trial-and-error and input optimization techniques that were suggested by earlier studies.

Table 4. Parametric summary of adopted techniques.

ML Technique	Parameters					
SDT	Min. rows in a node 5	Min. size node to split 10	Max. tree levels 10	Cross-validation trees 10	Smooth min. spikes 3	- -
TB	Max. trees in series 400	Min. size node to split 10	Trimming factor 0.01	Min. trees in series 10	Smooth min. spikes 5	Random percent 20
DTF	Trees in forest 200	Min. size node to split 2	Max. tree levels 50	Random predictor control square root of total predictors		- -
MLP	Hidden layers 1	Min. neurons 2	Max. neurons 20	Cross-validation V-fold	Max. iteration 10,000	Alg. LV-Marquardt
GEP	Population size 50	Max. initial pop. 10,000	Max. generation 2000	Gene per chromosome 4	Gene head length 8	Fitness threshold 1

2.9. Model Performance Evaluation

The effectiveness of the models established for rainfall–runoff modelling in this research may be determined using a variety of analytical methods. The variety of computational models used in research, as well as their results, strongly affects the choice of performance assessment indicators. To assess the effectiveness of our developed models, we used a combination of empirical measures, such as the coefficient of determination (R^2), root mean squared error (RMSE), and Nash–Sutcliffe efficiency (NSE) [65]. These statistical measures can be calculated using following equations.

$$R^2 = \frac{[n \sum_{i=1}^n (Q_{obs} \times Q_{mod}) - (\sum_{i=1}^n Q_{obs}) (\sum_{i=1}^n Q_{mod})]^2}{[n \sum_{i=1}^n Q_{obs}^2 - (Q_{mod})^2] - [n \sum_{i=1}^n Q_{mod}^2 - (Q_{mod})^2]} \quad (8)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Q_{obs} - Q_{mod})^2} \quad (9)$$

$$NSE = 1 - \frac{\sum_{i=1}^n (Q_{obs} - Q_{mod})^2}{\sum_{i=1}^n (Q_{obs} - \overline{Q_{obs}})^2} \quad (10)$$

where Q_{obs} and Q_{mod} denote the observed and modelled runoff using the developed models, respectively, while $\overline{Q_{obs}}$ is the average of observed runoff, and n is number of observations.

Other performance evaluation measures include mean absolute error (MAE), mean absolute percentage error (MAPE), mean percentage error (MPE), Kling–Gupta efficiency (KGE), etc. The selection of NSE, RMSE, and R^2 is done to cover all three aspects of accuracy metrics (correlation, error, and efficiency) and to avoid repetitions of the evaluation measures.

3. Results and Discussion

3.1. Single Decision Tree (SDT)

Initially, rainfall and runoff datasets were divided into 70% training set (5114 observations) and 30% testing set (1826 observations) each. Then the rainfall training set was selected as the predictor variable while the runoff training set was selected as the target variable. The initial model of SDT was generated without any incorporation of time lags in input datasets. In the following iterations, lag order (Lo) of input was gradually increased

from 0 to 10, as shown in Table 3; thus, a total of 11 SDT with each having distinct input (i.e., Lo0, Lo1, Lo2, . . . , Lo10) were created. Likewise, in further analysis, each input was pre-processed using the MODWT technique to assess the effect of wavelet transformation on SDT models. The parametric details and model formation of MODWT models is given in relevant sections.

It can be observed from Figure 8 that simple SDT models outperformed the MODWT-processed SDT model at lower lags (from Lo0 to Lo4), whereas at higher lag order, i.e., from Lo5 to Lo10, MODWT-SDT models performed better than SDT models. Highest testing *NSE* was observed for the SDT—Lo4 model, which was equal to 0.22. Contrastingly, lowest the *RMSE* was observed for MODWT-SDT at Lo6, which was equal to 65.86 m³/s. Average *NSE* for SDT models in training was 0.12 and for testing, it was 0.11. Meanwhile, MODWT-SDT models yielded 0.29 *NSE* in training and 0.13 in testing. In conclusion, neither the SDT nor the MODWT-SDT models could yield acceptable or satisfactory results.

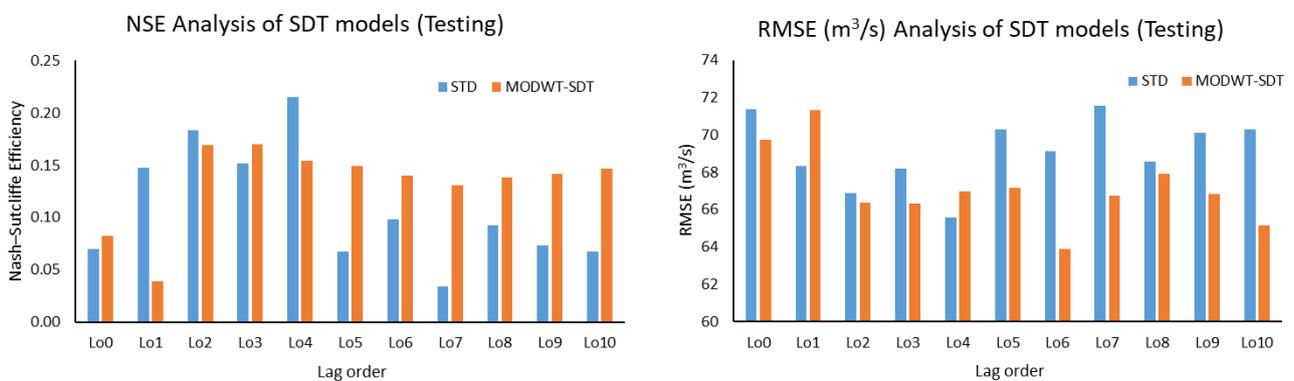


Figure 8. Performance evaluation of the SDT and MODWT-SDT models.

3.2. Decision Tree Forest (DTF)

After SDT analysis, DTF was applied in a similar parametric setting. DTF models and MODWT-DTF models were formed by using time lagged inputs with and without wavelet pre-processing. Outcomes of each model were statistically assessed through the performance evaluation measures (Figure 9).

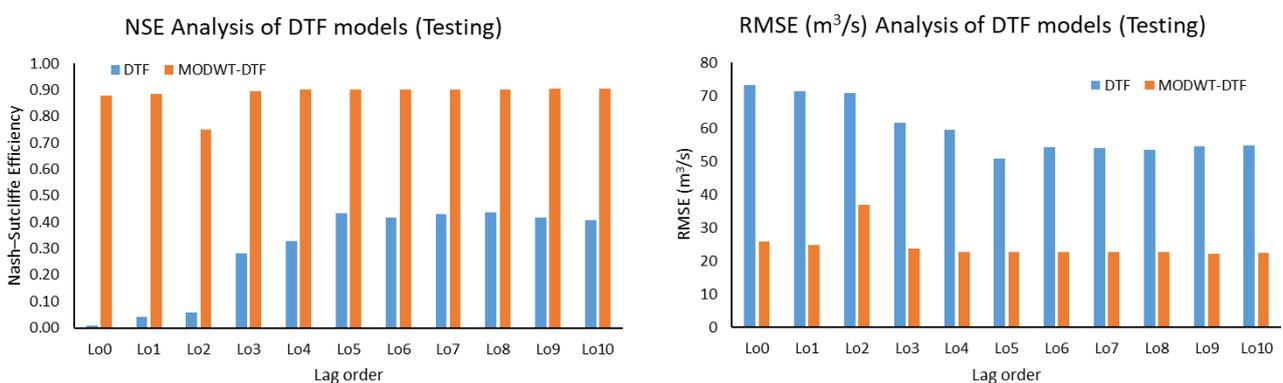


Figure 9. Performance evaluation of the DTF and MODWT-DTF models.

Performance of DTF models is graphically expressed in Figure 9, where it can be noted that MODWT-DTF models significantly outperformed DTF models. Average *NSE* in testing for MODWT-DTF was observed to be 0.88, while maximum efficiency was observed at Lo9. However, owing to the fact that efficiency marginally increased from 0.90 at Lo3 to 0.91 at Lo9, it can be considered that MODWT-DTF yielded best performance at Lo3. While average *NSE*, for testing with DTF models, was limited at a low value of 0.30, and maximum *NSE* for simple models was observed at Lo8, which was equal to 0.44. Similarly,

MODWT-DTF models yielded lower *RMSE* values than DTF models. Average *RMSE* for MODWT-DTF in testing was $22.84 \text{ m}^3/\text{s}$, while for DTF models, it was $59.96 \text{ m}^3/\text{s}$. In conclusion, wavelet pre-processing significantly improved DTF performance and enhanced testing *NSE* by a factor of 2.98 on average, while in training average *NSE* improved from 0.41 to 0.90 by a factor of 2.24.

3.3. Tree Boost (TB)

Figure 10 shows the performance of TB models in terms of *NSE* and *RMSE* (m^3/s). Here, it can be observed that MODWT-TB models performed well as compared to simple TB models. Maximum performance efficiency in testing was 0.17 at Lo3 for the MODWT-TB model. Additionally, minimum *RMSE* was observed at Lo3 for the MODWT-TB model, which was equal to $66.42 \text{ m}^3/\text{s}$, whereas average *NSE* of simple TB models in testing was only 0.04 as compared to 0.11 for MODWT-TB models. Likewise, lower *RMSE* values were yielded by MODWT-TB as compared to TB models, as average *RMSE* for TB was $71.59 \text{ m}^3/\text{s}$ in training and $71.91 \text{ m}^3/\text{s}$ in testing, while for MODWT-TB, average *RMSE* in training was 61.93 and $68.54 \text{ m}^3/\text{s}$ in testing. Although wavelet transformation improved the performance of simple TB models, but still these models did not perform well in comparison to other techniques.

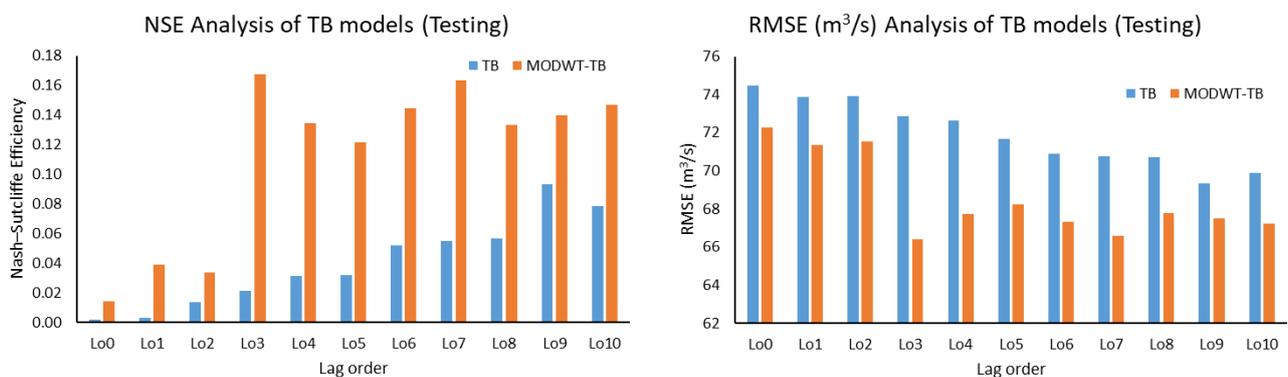


Figure 10. Performance evaluation of the TB and MODWT-TB models.

3.4. Multilayer Perceptron (MLP)

Similarly, the performance of MLP and MODWT-MLP models as per evaluation criterion is shown in Figure 11. The graphs indicate that MODWT-MLP models performed better than simple MLP, as supported by *NSE* and *RMSE* values. The former produced better *NSE* at all lag orders and also with lower *RMSE* values. The highest *NSE* value of 0.33 was observed at Lo4 for the MODWT-MLP model, which also produced the lowest *RMSE* value of $60.57 \text{ m}^3/\text{s}$. The average *NSE* for MLP models in training was 0.27 and testing was 0.20, while for MODWT-MLP models, these values were 0.32 and 0.28, respectively. Similarly, average *RMSE* for MLP in training and testing was 62.33 and $65.67 \text{ m}^3/\text{s}$ respectively, as compared with 60.62 and $62.53 \text{ m}^3/\text{s}$ for MODWT-MLP models. In conclusion, the best performance was observed at Lo4 in this case. Moreover, wavelet pre-processing enhanced the performance of MLP models, but still it was not satisfactory.

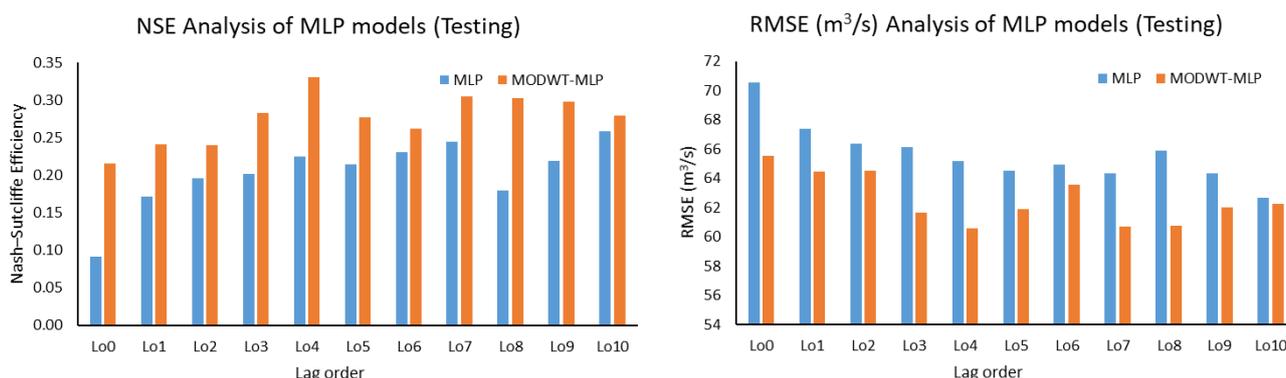


Figure 11. Performance evaluation of the MLP and MODWT-MLP models.

3.5. Gene Expression Programming (GEP)

In Figure 12, performance of the GEP and MODWT-GEP models is graphically presented, which clearly indicates that MODWT-GEP models outperformed simple GEP models. Wavelet pre-processed GEP models produced runoff simulation with higher accuracy than that of simple GEP models, as average *NSE*, in testing, for the former was 0.26 as compared to 0.18 of the later. Additionally, MODWT-GEP produced a lower testing *RMSE*, on average, i.e., 62.91 m³/s, as compared to 66.57 m³/s for MLP models. The highest *NSE* value in testing was observed at Lo6 for MODWT-MLP, equal to 0.33, while a minimum *RMSE* of 60.85 m³/s was observed at the same lag order for the same model. Similar to TB, MLP, and SDT, gene expression programming also produced unsatisfactory results, although wavelet transformation indicated signs of accuracy enhancement.

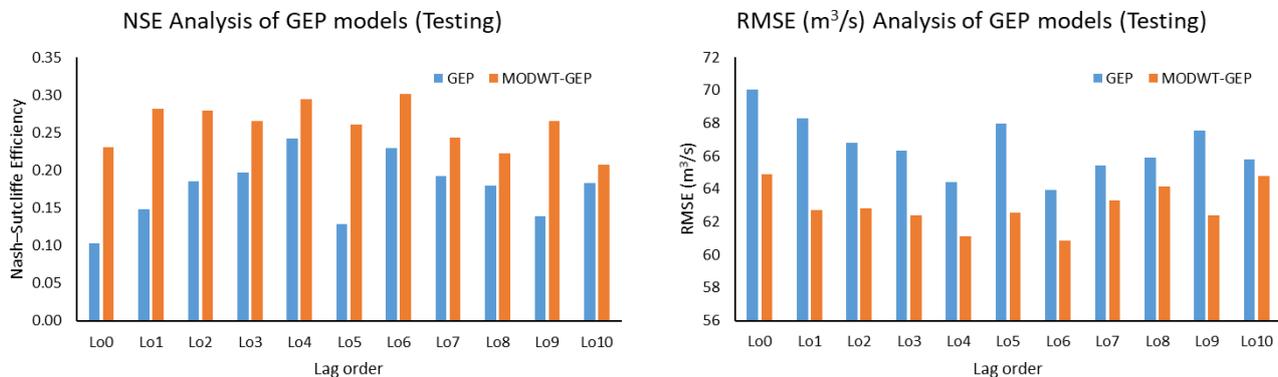


Figure 12. Performance evaluation of the GEP and MODWT-GEP models.

3.6. Comparative Analysis

After a thorough performance evaluation of each technique with and without wavelet transformation, an encompassing comparative analysis of best performing models of each technique is described in this section. For the comprehensive comparative analysis, from each technique used in this study, one model with best performance indices is selected. Table 5 summarizes the statistical metrics and wavelet parameters of the models. It can be noted that, in each technique except single direction tree, simple models integrated with wavelet pre-processing outperformed the countermodels having pre-processed inputs. Moreover, the analysis shows that these models mostly performed well at lower lag orders, i.e., Lo3 and Lo4, except MODWT-GEP, which performed best at Lo6. Further analysis indicated that wavelet-coupled decision tree forest (MODWT-DTF) outperformed all other models in this study. The highest testing *NSE* value (0.90) for rainfall–runoff modelling of the Soan River basin was produced at lag order 4 (Lo4) by MODWT-DTF. This value accounts for 90% modelling accuracy along with minimum modelling error in terms of *RMSE* equal to 23.79 m³/s. Figure 13 graphically represents the comparison of models

with highest accuracy in each technique. Moreover, we also compared the best performing model with respect to their accuracy in modelling flows of higher magnitude along with lower magnitude. For this purpose, the performance was evaluated at runoff below the third quartile (Q3) for low flows and above Q3 for high flows, whereas the value of Q3 for the testing data was 38.74 m³/s. For the comparative analysis, it was observed that MODWT-DTF at Lo3 performed best in both scenarios, i.e., low-magnitude runoff and high-magnitude runoff. However, the *NSE* of the model dropped slightly from 0.90 above Q3 to 0.85 below Q3, whereas the *R*² value dropped significantly from 0.94 to 0.55 from high to low magnitude runoff. Interestingly, MODWT-TB at Lo3 yielded a high *NSE* value of 0.79 for low-magnitude runoff, but the model produced a higher *RMSE* value and lower *R*² and *NSE* values compared to MODWT-DTF at Lo3 (Table 6). Therefore, the latter model is considered to be outperforming the former in both scenarios.

Table 5. Summary of best performing model comparison.

ML Technique	Lag Order	Wavelet Filter	Level	Testing		
				<i>NSE</i>	<i>RMSE</i>	<i>R</i> ²
SDT	Lo4	-		0.22	65.52	0.22
MODWT-DTF	Lo3	db4	3	0.90	23.79	0.94
MODWT-TB	Lo3	db4	3	0.17	66.42	0.24
MODWT-MLP	Lo4	db4	3	0.33	60.57	0.33
MODWT-GEP	Lo6	db4	3	0.30	60.85	0.31

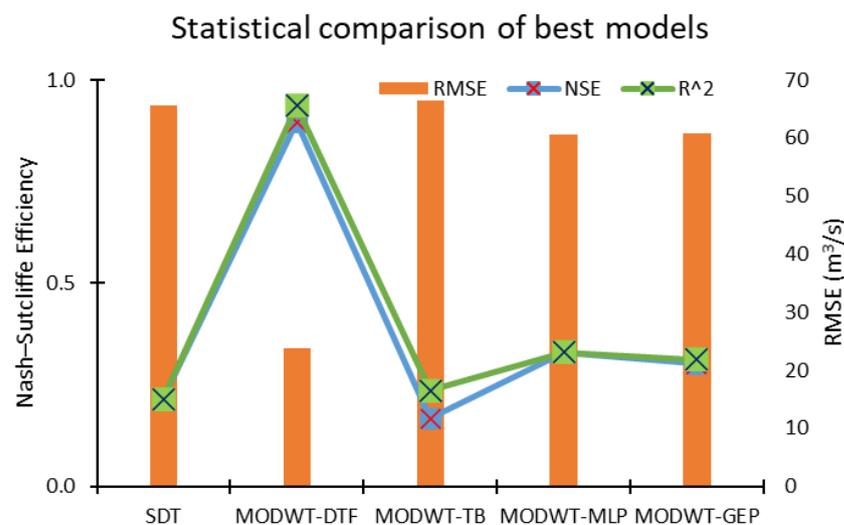


Figure 13. Comparative analysis of the best models.

Table 6. Summary of best performance models at low flows and high flows.

ML Technique	Performance below Q3			Performance above Q3		
	<i>NSE</i>	<i>RMSE</i>	<i>R</i> ²	<i>NSE</i>	<i>RMSE</i>	<i>R</i> ²
SDT	0.04	22.23	0.05	0.31	57.49	0.28
MODWT-DTF	0.85	8.53	0.55	0.90	21.54	0.94
MODWT-TB	0.79	10.12	0.08	0.11	65.65	0.16
MODWT-MLP	0.28	18.54	0.12	0.37	54.99	0.29
MODWT-GEP	0.28	18.56	0.17	0.30	57.95	0.21

As previously discussed, MODWT-MLP produced the highest modelling accuracy, 90% at Lo4; Figure 14 shows the rainfall–runoff plot of the model compared with observed runoff. It can be observed that MODWT-MLP-Lo4 (solid orange line) replicated the observed runoff (solid black line) with high accuracy. The model successfully captured the

trends of observed runoff at extreme values (outliers) along with recurring frequencies. Figure 15 shows the regression plot of the models, which indicates a high coefficient of determination (R^2) value equal to 0.9405, which is also presented in Table 5.

Rainfall–Runoff Modelling of Soan River basin (MODWT-DTF-Lo3)

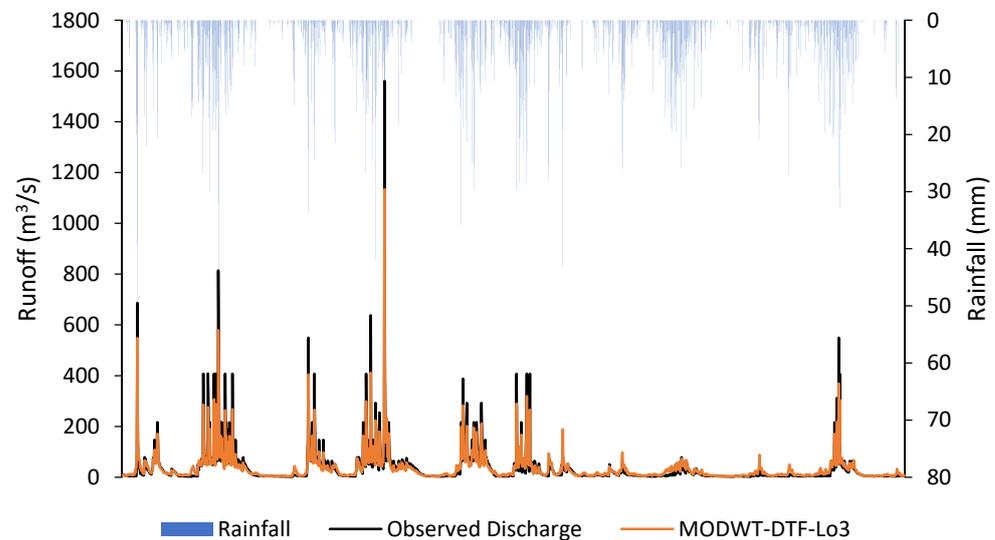


Figure 14. Rainfall–runoff modelling of Soan River basin (MODWT-DTF-Lo3).

Regression Plot (MODWT-DTF-Lo3)

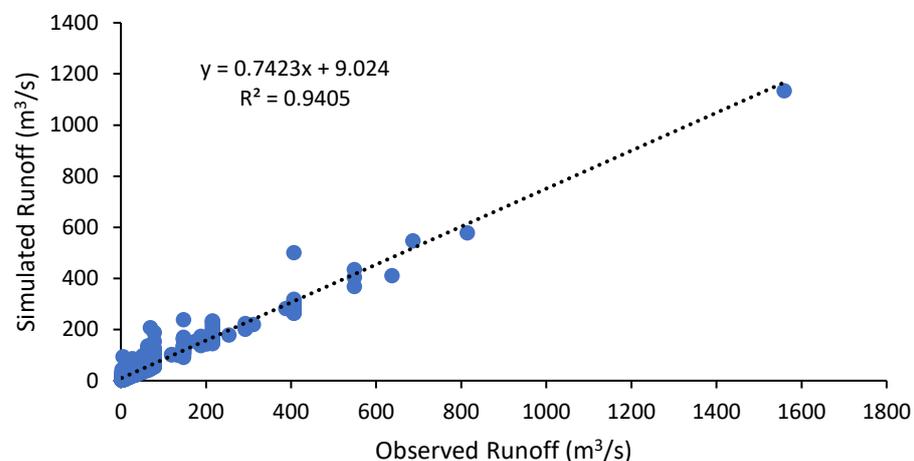


Figure 15. Regression plot (MODWT-DTF-Lo3).

3.7. Discussion

In this study, we used major machine learning techniques together with MODWT pre-processing to model the rainfall–runoff relationship in the Soan River basin. Outcomes suggested that MODWT-based DTF performed best at lag order 3. The modelling accuracy was observed to be 90% with RMSE value as low as 23.79 m³/s. Moreover, it was also observed that wavelet pre-processing enhanced the effectiveness of simple models. An overarching research work was done regarding application of machine learning techniques in hydrological problems. This study adds to the literature by developing an encompassing comparison of the performance of five mainstream machine learning techniques, along with assessing the effect of MODWT pre-processing, in rainfall–runoff modelling of the Soan River basin, the largest basin in the Pothohar plateau.

Machine learning techniques have been intensively applied to hydrological problems by undertaking real-world case studies. Wu and Chau [66] used a different combination of two signal pre-processing techniques, singular spectrum analysis (SSA) and moving average (MA), with two ML techniques, ANN and support vector regression, to forecast rainfall with multiple lead times. Using a similar approach, this current study separately combined five ML techniques with MODWT pre-processing method to explore the rainfall–runoff relationship in the Soan River basin. Similarly, the impact of changing climate on the occurrence of extreme hydrological events in an Iranian catchment was assessed by Sharafati and Elnaz [67]. In contrast to our study, the authors used Soil and Water Assessment Tool (SWAT) and a combination of global climate models to study rainfall and runoff trends in the catchment. Similar to this study, Ghani et al. [68] also mapped the rainfall–runoff association in the Pothohar region and Soan River basin with 92% accuracy using HEC-HMS and HEC-GeoHMS software. Likewise, Hussain et al. [69] also modelled the rainfall–runoff relation of the region with maximum R^2 value of 0.66. Recently, Ouma et al. [70] also modelled the rainfall–runoff relationship in the Nzoia subbasin of Lake Victoria using wavelet-based machine learning models and compared their performance with the long short-term memory (LSTM) deep learning model. Outcomes of our study provide empirical support to their findings corresponding to performance enhancement of machine learning models when coupled with wavelet pre-processing. The optimal lag order for the Soan basin was observed to be three, i.e., the models performed best when rainfall data were lagged up to three days having input coefficients R_{t-1} , R_{t-2} , and R_{t-3} . This optimal lag number related to the time in which runoff generated from rainfall at the farthest point of the basin reaches the outlet. From analysis of the rainfall and runoff training data, it was observed that highest runoff ($1407.69 \text{ m}^3/\text{s}$) was generated after two days of highest rainfall incident (95.48 mm), whereas similar observation was made during the testing period when the highest runoff was observed after two days and lasted up to three days of the maximum rainfall event. This association between theoretical concept of lag order (also called time lag or lag number) and physical concept (time of concentration) has also been observed by earlier studies [37,40,71].

This study added to the literature by indicating the effectiveness of DTF and MODWT-based DTF applied in the Soan River basin modelling; whereas, various previous studies solidify the findings of our study regarding accuracy enhancement by wavelet transformation [25,46,48].

3.8. Research Limitations and Uncertainties

Although the Soan River basin is the largest in the Pothohar region, it does not fully represent the region, as there exist five other basins. To fully understand the dynamics of rainfall and runoff association in the region, further research work is needed on different basins of the region. Moreover, application of deep learning techniques in hydrology has emerged progressively; therefore, deep learning techniques such as long short-term memory (LSTM) might be helpful in exploring new dimensions of the basin's hydrological characteristics. This research only uses rainfall as an output and pre-processed rainfall data to extract additional and hidden information from the time series, but combinations of rainfall, infiltration, and evaporation can also be used in any subsequent study to better understand the relationship of these hydrological processes. Moreover, other techniques of signal processing, such as bootstrapping, can also be used to assess their impact on the performance these models. These recommendations for future research work are most likely to be undertaken by the authors in the near future.

4. Conclusions

This study used five machine learning techniques (SDT, TB, DTF, MLP, and GEP) with and without combination of maximal overlap discrete wavelet transformation (MODWT). From the comparative analysis of the simple and MODWT-based models, it can be concluded that MODWT-based models outperformed simple models, except SDT, where the

simple SDT model yielded higher accuracy than MODWT-SDT models. Moreover, it was observed that MODWT-SDT performed better than SDT models at higher lag orders. Additionally, it was observed that MODWT pre-processing improved the performance of simple models, but it did not necessarily push the accuracy within acceptable or satisfactory range, whereas the comparative analysis of the best performing models from technique established that DTF outperformed the other four techniques, when coupled with MODWT. Highest modelling accuracy, equal to 90%, was produced at MODWT-DTF at Lo3. Meanwhile, the same model produced a minimum *RMSE* of 23.79 m³/s. This performance enhancement reflects the tendency of data pre-processing with MODWT (or any other wavelet model) to extract hidden information within the univariate time series. Hence, wavelet pre-processing can be reliably applied to problems involving multiple variables to attain better outcomes with reduced parametric requirement and parsimonious modelling. However, the results of wavelet technique can also change depending on the type, topography, and features of the basin; therefore, selection of wavelet family, wavelet type, and level of decomposition must be done correctly to achieve better outcomes.

Author Contributions: Conceptualization, M.S., M.T.K. and F.A.; methodology, M.T.K., M.H. and H.S.; resources, M.T.K. and F.A.; data curation, M.T.K.; formal analysis, H.S. and M.T.K.; writing and original draft preparation, M.H.; writing—review and editing, M.S., S.A. and M.H.; supervision, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Higher Education Commission (HEC) of Pakistan (No. 7368).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the study was obtained from Pakistan Meteorological Department (PMD), WAPDA and Soil & Water Conservation Research Institute Chakwal (SAWCRI).

Acknowledgments: The authors are grateful to Pakistan Meteorological Department (PMD), WAPDA and Soil & Water Conservation Research Institute Chakwal (SAWCRI) for providing the data used in this study.

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

References

1. Chang, T.K.; Talei, A.; Alaghmand, S.; Ooi, M. 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](#)]
2. Tokar, A.S.; Johnson, P.A. Rainfall-Runoff Modeling Using Artificial Neural Networks. *J. Hydrol. Eng.* **1999**, *4*, 232–239. [[CrossRef](#)]
3. Tikhamarine, Y.; Malik, A.; Kumar, A.; Souag-Gamane, D.; Kisi, O. Estimation of monthly reference evapotranspiration using novel hybrid machine learning approaches. *Hydrol. Sci. J.* **2019**, *64*, 1824–1842. [[CrossRef](#)]
4. Himmelblau, D.M. *Basic Principles and Calculations in Chemical Engineering*, 2nd ed.; Prentice Hall: Hoboken, NJ, USA, 1967.
5. Bartoletti, N.; Casagli, F.; Marsili-Libelli, S.; Nardi, A.; Palandri, L. Data-driven rainfall/runoff modelling based on a neuro-fuzzy inference system. *Environ. Model. Softw.* **2018**, *106*, 35–47. [[CrossRef](#)]
6. Peel, M.C.; McMahon, T.A. Historical development of rainfall-runoff modeling. *WIREs Water* **2020**, *7*, e1471. [[CrossRef](#)]
7. Mulvaney, T.J. On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and flood discharges in a given catchment. *Trans. Inst. Civil Eng. Irel.* **1851**, *4*, 19–31.
8. Schreiber, P. About the relationship between precipitation and the flow of water in rivers in Central Europe. *Z. Meteorol.* **1904**, *21*, 441–452. (In German)
9. Sherman, L.K. Streamflow from Rainfall by the Unit-graph Method. *Eng. News Rec.* **1932**, *108*, 501–505.
10. Nash, J.E. The form of the Instantaneous Unit Hydrograph. *Int. Assoc. Sci. Hydrol.* **1957**, *3*, 114–121.
11. Linsley, R.K.; Crawford, N.H. Computation of a synthesis streamflow record on a digital computer. *Int. Assoc. Sci. Hydrol.* **1960**, *5*, 526–538.
12. Huggins, L.F.; Monke, E.J. A Mathematical Model for Simulating the Hydrologic Response of a Watershed. *Water Resour. Res.* **1968**, *4*, 529–539. [[CrossRef](#)]
13. Daniell, T.M. Neural Networks—Applications in hydrology and water resources engineering. *Natl. Conf. Publ. Proc. Inst. Eng.* **1991**, 797–902.

14. French, M.N.; Krajewski, W.; Cuykendall, R.R. Rainfall forecasting in space and time using a neural network. *J. Hydrol.* **1992**, *137*, 1–31. [[CrossRef](#)]
15. Asadi, H.; Shahedi, K.; Jarihani, B.; Sidle, R.C. Rainfall-Runoff Modelling Using Hydrological Connectivity Index and Artificial Neural Network Approach. *Water* **2019**, *11*, 212. [[CrossRef](#)]
16. Dawson, C.; Wilby, R. An artificial neural network approach to rainfall-runoff modelling. *Hydrol. Sci. J.* **1998**, *43*, 47–66. [[CrossRef](#)]
17. Kumar, S.; Roshni, T.; Himayoun, D. A Comparison of Emotional Neural Network (ENN) and Artificial Neural Network (ANN) Approach for Rainfall-Runoff Modelling. *Civ. Eng. J.* **2019**, *5*, 2120–2130. [[CrossRef](#)]
18. Poonia, V.; Tiwari, H.L. Rainfall-runoff modeling for the Hoshangabad Basin of Narmada River using artificial neural network. *Arab. J. Geosci.* **2020**, *13*, 944. [[CrossRef](#)]
19. Shamseldin, A.Y.; Nasr, A.E.; Oconnor, K.M. Comparison of different forms of the Multi-layer Feed-Forward Neural Network method used for river flow forecasting. *Hydrol. Earth Syst. Sci.* **2002**, *6*, 671–684. [[CrossRef](#)]
20. Sharghi, E.; Nourani, V.; Najafi, H.; Molajou, A. Emotional ANN (EANN) and Wavelet-ANN (WANN) Approaches for Markovian and Seasonal Based Modeling of Rainfall-Runoff Process. *Water Resour. Manag.* **2018**, *32*, 3441–3456. [[CrossRef](#)]
21. 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](#)]
22. Shoaib, M.; Shamseldin, A.; Melville, B.; Khan, M.M. A comparison between wavelet based static and dynamic neural network approaches for runoff prediction. *J. Hydrol.* **2016**, *535*, 211–225. [[CrossRef](#)]
23. Abbot, J.; Marohasy, J. Application of artificial neural networks to rainfall forecasting in Queensland, Australia. *Adv. Atmos. Sci.* **2012**, *29*, 717–730. [[CrossRef](#)]
24. Gomes, E.P.; Blanco, C.J.C. Daily rainfall estimates considering seasonality from a MODWT-ANN hybrid model. *J. Hydrol. Hydromech.* **2021**, *69*, 13–28. [[CrossRef](#)]
25. Hammad, M.; Shoaib, M.; Salahudin, H.; Baig, M.A.I.; Khan, M.M.; Ullah, M.K. Rainfall forecasting in upper Indus basin using various artificial intelligence techniques. *Stoch. Environ. Res. Risk Assess.* **2021**, *35*, 2213–2235. [[CrossRef](#)]
26. Lee, J.; Kim, C.-G.; Lee, J.E.; Kim, N.W.; Kim, H. Application of Artificial Neural Networks to Rainfall Forecasting in the Geum River Basin, Korea. *Water* **2018**, *10*, 1448. [[CrossRef](#)]
27. Lin, G.-F.; Wu, M.-C. A hybrid neural network model for typhoon-rainfall forecasting. *J. Hydrol.* **2009**, *375*, 450–458. [[CrossRef](#)]
28. 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](#)]
29. Toth, E.; Brath, A.; Montanari, A. Comparison of short-term rainfall prediction models for real-time flood forecasting. *J. Hydrol.* **2000**, *239*, 132–147. [[CrossRef](#)]
30. Adnan, R.M.; Petroselli, A.; Heddam, S.; Santos, C.A.G.; Kisi, O. Short term rainfall-runoff modelling using several machine learning methods and a conceptual event-based model. *Stoch. Environ. Res. Risk Assess.* **2020**, *35*, 597–616. [[CrossRef](#)]
31. Mehr, A.D.; Nourani, V. Season Algorithm-Multigene Genetic Programming: A New Approach for Rainfall-Runoff Modelling. *Water Resour. Manag.* **2018**, *32*, 2665–2679. [[CrossRef](#)]
32. Nourani, V.; Tajbakhsh, A.D.; Molajou, A. Data mining based on wavelet and decision tree for rainfall-runoff simulation. *Hydrol. Res.* **2018**, *50*, 75–84. [[CrossRef](#)]
33. Sharghi, E.; Nourani, V.; Molajou, A.; Najafi, H. Conjunction of emotional ANN (EANN) and wavelet transform for rainfall-runoff modeling. *J. Hydroinform.* **2018**, *21*, 136–152. [[CrossRef](#)]
34. Aytek, A.; Asce, M.; Alp, M. An application of artificial intelligence for rainfall-runoff modeling. *J. Earth Syst. Sci.* **2008**, *117*, 145–155. [[CrossRef](#)]
35. Nourani, V.; Tajbakhsh, A.D.; Molajou, A.; Gokcekus, H. Hybrid Wavelet-M5 Model Tree for Rainfall-Runoff Modeling. *J. Hydrol. Eng.* **2019**, *24*, 04019012. [[CrossRef](#)]
36. 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](#)]
37. Raza, A.; Shoaib, M.; Khan, A.; Baig, F.; Faiz, M.A.; Khan, M.M. Application of non-conventional soft computing approaches for estimation of reference evapotranspiration in various climatic regions. *Theor. Appl. Clim.* **2019**, *139*, 1459–1477. [[CrossRef](#)]
38. Allen, R.G.; Pereira, L.S.; Raes, D.; Smith, M. Guidelines for Computing Crop Water Requiriments, Rome. 1998. Available online: <http://www.kimberly.uidaho.edu/water/fao56/fao56.pdf%5Cnhttp://linkinghub.elsevier.com/retrieve/pii/S1161030110001103> (accessed on 24 August 2020).
39. Shoaib, M.; Shamseldin, A.Y.; Melville, B.W. Hybrid Wavelet Neural Network Approach. In *Network Modelling, Studies in Computational Intelligence*; Springer: Cham, Switzerland, 2016; pp. 127–143.
40. Shoaib, M.; Shamseldin, A.Y.; Khan, S.; Sultan, M.; Ahmad, F.; Sultan, T.; Dahri, Z.H.; Ali, I. Input Selection of Wavelet-Coupled Neural Network Models for Rainfall-Runoff Modelling. *Water Resour. Manag.* **2018**, *33*, 955–973. [[CrossRef](#)]
41. Nabi, G.; Hussain, F.; Wu, R.-S.; Nangia, V.; Bibi, R. Micro-Watershed Management for Erosion Control Using Soil and Water Conservation Structures and SWAT Modeling. *Water* **2020**, *12*, 1439. [[CrossRef](#)]
42. Hussain, F.; Nabi, G.; Wu, R.-S. Spatiotemporal Rainfall Distribution of Soan River Basin, Pothwar Region, Pakistan. *Adv. Meteorol.* **2021**, *2021*, 6656732. [[CrossRef](#)]
43. Ashraf, A. Risk modeling of soil erosion under different land use and rainfall conditions in Soan river basin, sub-Himalayan region and mitigation options. *Model. Earth Syst. Environ.* **2019**, *6*, 417–428. [[CrossRef](#)]

44. Hussain, F. *Impact of Rainfall and Landuse Pattern on Water and Sediment Yield in Soan Basin*; University of Engineering and Technology: Lahore, Pakistan, 2016.
45. Stone, M. Cross-Validatory Choice and Assessment of Statistical Predictions. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1974**, *36*, 111–133. [[CrossRef](#)]
46. Adamowski, J.; Sun, K. Development of a coupled wavelet transform and neural network method for flow forecasting of non-perennial rivers in semi-arid watersheds. *J. Hydrol.* **2010**, *390*, 85–91. [[CrossRef](#)]
47. 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](#)]
48. Quilty, J.; Adamowski, J. Addressing the incorrect usage of wavelet-based hydrological and water resources forecasting models for real-world applications with best practices and a new forecasting framework. *J. Hydrol.* **2018**, *563*, 336–353. [[CrossRef](#)]
49. Kotsiantis, S.B. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* **2007**, *31*, 249–268.
50. Breiman, L. *Arcing the Edge*; Technical Report 486; University of California: Berkeley, CA, USA, 1998.
51. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
52. Webros, P.J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974.
53. Goodfellow, Y.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
54. Bishop, M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
55. Babs, T. The Mathematics of Neural Networks. 2018. Available online: <https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05> (accessed on 24 August 2020).
56. Ferreira, C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Syst.* **2001**, *13*, 87–129.
57. Grossmann, A.; Morlet, J. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.* **2009**, *15*, 126–139. [[CrossRef](#)]
58. Walden, A.T. Wavelet Analysis of Discrete Time Series. In *Progress in Mathematics*; Springer: Berlin/Heidelberg, Germany; Volume 202, pp. 627–641.
59. Percival, B.; Walden, A.T. The Maximal Overlap Discrete Wavelet Transform. In *Wavelet Methods for Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2000; pp. 159–205.
60. 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](#)]
61. Bracewell, R.N.; Bracewell, R.N. *The Fourier Transform and Its Applications*; McGraw-Hill: New York, NY, USA, 1986; Volume 31999.
62. May, R.J.; Dandy, G.C.; Maier, H.; Nixon, J.B. Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. *Environ. Model. Softw.* **2008**, *23*, 1289–1299. [[CrossRef](#)]
63. Ribeiro, G.H.T.; Neto, P.S.G.D.M.; Cavalcanti, G.D.C.; Tsang, I.R. Lag selection for time series forecasting using Particle Swarm Optimization. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2437–2444. [[CrossRef](#)]
64. Maheswaran, R.; Khosa, R. Wavelet-Volterra coupled model for monthly stream flow forecasting. *J. Hydrol.* **2012**, *450–451*, 320–335. [[CrossRef](#)]
65. Mills, G. Ireland’s water budget—Model validation and a greenhouse experiment. *Ir. Geogr.* **2001**, *34*, 124–134. [[CrossRef](#)]
66. Wu, C.; Chau, K.W. Prediction of rainfall time series using modular soft computing methods. *Eng. Appl. Artif. Intell.* **2013**, *26*, 997–1007. [[CrossRef](#)]
67. Sharafati, A.; Pezeshki, E. A strategy to assess the uncertainty of a climate change impact on extreme hydrological events in the semi-arid Dehbar catchment in Iran. *Theor. Appl. Clim.* **2019**, *139*, 389–402. [[CrossRef](#)]
68. Ghani, M.W.; Arshad, M.; Shabbir, A.; Shakoor, A.; Mehmood, N.; Ahmad, I. Investigation of potential water harvesting sites at potohar using modeling approach. *Pakistan J. Agric. Sci.* **2013**, *50*, 723–729.
69. Hussain, F.; Wu, R.-S.; Nabi, G.; Hussain, B. A Study on Rainfall, Runoff and Sediment Yield Relations in Small Watersheds of Pothwar Region Pakistan. In Proceedings of the 2017 Conference on Sustainable Development for Environmental Resources, Zhongli, Taiwan, December 2017.
70. Ouma, Y.O.; Cheruyot, R.; Wachera, A.N. Rainfall and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: Case study of Nzoia hydrologic basin. *Complex Intell. Syst.* **2021**, 1–24. [[CrossRef](#)]
71. 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](#)]