ORIGINAL ARTICLE

Artificial intelligence for suspended sediment load prediction: a review

Deepak Gupta[1] · Barenya Bikash Hazarika[1] · Mohanadhas Berlin[2] · Usha Mary Sharma[1] · Kshitij Mishra[1]

Abstract

The estimation of sediment yield concentration is crucial for the development of stream ventures, watershed management, toxins estimation, soil disintegration, floods, and so on. In this study, we summarize various existing artificial intelligence (AI)-based suspended sediment load (SSL) estimation models to calculate the suspended sediment load, to our knowledge to date. The artificial neural network (ANN), generalized regression neural network (GRNN), neuro-fuzzy (NF), genetic algorithm (GA), gene expression programming (GEP), classification and regression tree (CART), linear regression (LR), multilinear regression (MLR), Chi-squared automatic interaction detection (CHAID), extreme learning machine (ELM), and support vector machine (SVM) are among the many AI-based models that have been successfully implemented for sediment load prediction. In this paper, we describe a few popular AI-based models that have been used for SSL prediction. ANN, SVM, and NF had overcome each other in different circumstances of prediction; and all three can be said as good predictors. Models using ANN with ELM or wavelet analysis in some ways are good predictors as their predicted values generally lie closer to the measured value. Performances of the algorithms are usually evaluated by applying various types of performance assessment methods most commonly RMSE, R^2, MAE, etc. This review is required to bear some significance to the researchers and hydrologists while seeking models that have been effectively actualized inSSLestimation or in hydrology related aspects, however, mainly focused on the researches between January 2015 and November 2020.

Keywords Suspended sediment load · Prediction · Artificial intelligence · Machine learning · Hydrology

✉ Deepak Gupta
deepakjnu85@gmail.com; deepak@nitap.ac.in

Barenya Bikash Hazarika
barenya1431@gmail.com

Mohanadhas Berlin
berlin@nitap.ac.in

Usha Mary Sharma
ushamary.sharma@gmail.com

Kshitij Mishra
kmishra.kings@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology, Yupia, Arunachal Pradesh 791112, India

[2] Department of Civil Engineering, National Institute of Technology, Yupia, Arunachal Pradesh 791112, India

Definition and basic concepts

Sediment transport is a burning question in river management practices. It shows great variation in sediment deposition throughout the river bed. The intense seasonal rainfall, streamflow, tropical climate, and immature geology are some of the factors which influence sediment transport and its deposition. Generally, sediment transport predominantly occurs during the monsoon season which results in a notable amount of sediment deposit the downstream of the river. Mostly, the sediments are in the form of earth materials and finally get flushed into the sea in the magnificent amount due to the sediment transport by the river. Naidu (1999) stated that 20 billion tons of earth materials on the planet get conveyed to the oceans every year by waterways or streams of which Indian subcontinent alone contributes for 6 billion, due to a large number of rivers and intense rainfall presence. In storm-water, the sediment load particles could accumulate on the top of soil surface or could be trapped underground soil pores. The transportation of

sediment frequently varies from one place to another place which affects greatly the process of sediment deposition. Hence, prediction of sediment load is essential for various civic development activities such as the dam designing, designing of reservoirs, watershed management and estimation of floods in flood-prone areas, etc. It is also essential to understand the sediment transport prediction during the development of Hydro-Power projects (Zarris et al. 2006, 2011). Hence, without a doubt, the precise estimation of suspended sediment load (SSL) plays a major part in hydraulic engineering as well as in civic development and river engineering practices(Brownlie 1981; Alonso et al., 1982).

Sediment transport mechanics is the study of fluid sediment motion laws and erosion, transport, and deposition processes. Various types of movement of sediments are found in nature, including the movement of sediments in rivers and canals, reservoirs, along the shore, and in the marine environment. The deserts and the pipelines are the results of flow, wind and waves of the stream. Stats indicate that 13 of the world's major rivers carry over 5.8 billion tones of annual sediment load (Chien and Wan 1998). There are peculiarities in a river that is heavily loaded with sediment that cause it to differ extensively from rivers that carry much less sediment. These differences have led to various engineering problems such as flood control, reservoir sedimentation, irrigation sedimentation of canals, and sedimentation problems in ports and estuaries (Duan and Takara 2020). According to Chien and Wan (1998), the mechanics of sediment transport should be a component of sediment science. In particular, this component should cover the following four aspects:

1. Sediment formation and its properties
2. Sediment transport mechanics
3. Field measurements and laboratory experiments
4. Applied science of sedimentation.

The sediment movement phenomenon is quite complicated. In general, sediment movement is a two-phase flow issue. Sediment moves under a flow's action, and its presence, in turn, influences the flow. In addition, practical problems arise when direct measurements are taken (Rezapour et al. 2010). Sediment transport is an intricate and non-linear process. Hence, it is a difficult task to model it (Kalteh et al., 2008). In the past, to perceive the mechanism for sediment transportation in rivers, great works have been enacted. As the evolution of river sediment science has taken place, the focus on sediment discharge estimation saw its growth. Sediment load in a river could be categorized into SSL and bed load (BL). The SSL corresponds to the major portion of the sediment load. BL depicts a particle in a flowing fluid that is transported along with the bed (Colby and Hembree 1955; Rijn 1984). A large number of researchers have studied the river SSL estimation and its simulation during the

last few decades. Direct measurements or indirect measurement through algorithms have been used to calculate the SSL of a stream. Direct measurements are directly carried on the site which has been selected for the experiment for gaining data, but it is uneconomical to acquire data at all locations and is not a feasible way with respect of time as it requires an enormous amount of time to collect satisfactory data. These direct measurements are more trustworthy than indirect measurements, but are avoided due to their complex nature. In this work, we have reduced the range of review models to the ones that specifically take account of SSL.

Since the suspended load prediction is a complex process; a comprehensive model is required for prediction, which will be accurate and easy to use. Sediment load is dependent on flow conditions as well as climatic conditions like rainfall, temperature (in some special cases), as well as on river delta mouth characteristics; hence, suspended sediment load prediction is a non-linear phenomenon to understand thoroughly, because it includes a number of interconnected components. It was found that the traditional models, viz., Einstein approach (Einstein 1950), Brook's approach (Brooks 1965), and SRC were used for suspended sediment load modeling (Kisi et al. 2006) before 1990. Furthermore, there was a tremendous turn of researchers toward the AI-based models like artificial neural network (ANN) (Tayfur and Gundal 2006) in various fields such as environmental engineering and water resource management. ANN algorithm is a very efficient and powerful computational machine learning algorithm utilized for simulating the complicated associations among variables which are non-linear (Gallantand Gallant 1993; Smithand Eli 1995; Yitianand Gu 2003). The application of ANN was performed in many areas other than river engineerings such as electrical engineering, image processing, financing, physics, neurophysiology, and others (Panagoulia et al. 2017).

In designing ANN models, some problems arise for high-value data and small value data; it does not provide satisfactory results in estimation compared to the actual value and converge to a local minimum. These ANN models for their better performance need a long duration training data, so that the over-fitting in the model could be avoided. To overcome these shortcomings, sometimes, it is inadequate to go for the ANN-based model. Hence, in this complex hydrological process, it would be better to use a tool which could provide a better solution to the problem taken. Vapnik and Cortes (1995) proposed a novel approach that uses the structural risk minimization principle, called SVM (Vapnik 1999, 2000). SVM is essentially implemented for solving problems concerned with classification and regression. The regression model is known as SVR (Drucker et al. 1997; Awad and Khanna 2015). They become popular because of their promising empirical performance. In several hydraulic engineering process and environmental problems, the SVM

is effectively implemented in recent decades (Flood and Kartam 1994; Sivapragasam et al. 2001; Dibike and Solomatine 2001; Sivapragasamand Muttil 2005; Tripathi et al. 2006; Lin et al.2006; Hong 2008; Khan and Coulibaly 2006; Chen and Li 2010; Yunkai et al. 2010; Noori et al. 2011; Ch et al. 2013; Ji and Lu 2018). SVM is used in suspended sediment estimation through its different models to estimate the SSL of two water bodies (Cimen 2008). Sediment yield simulation was also done through SVM by Misra et al. (2009). It was reported by them that as compared to ANN, SVM furnish better outcomes in training, testing, and validation. Azamathulla et al. (2010) in their work applied SVM to validate its predictive capability. They finally found that SVM displayed superior performance in comparison with the other traditional models. Whenever the outputs gathered through the usage of datasets, it was seen that SVM provided better results as compared to ANN for SSL estimation (Jie and Yu 2011). Hazarika et al. (2020a) compared the prediction performance of SVR and ANN model and discovered that SVR outperforms the ANN model. Hassanpour et al. (2019) showed the applicability of fuzzy C-mean clustering-based SVR model for suspended sediment load prediction. A variation of SVM is also used in modeling is known as least square SVM (LSSVM). LSSVM was introduced for demonstrating SSL relationship and it was discovered that the LSSVM model could over-play the ANN model and the two models executed superior to the SRC model (Kisi 2012). Lafdani et al. (2013) described the two models, viz., ANN and SVM through gamma test for input selection might prompt preferable effectiveness over the regression combination. For solving non-linear classification, LSSVM is a powerful methodology. Mondal (2011) proposed a new model, viz., gamma geomorphologic instantaneous unit hydrograph (GGIUH) for the estimation of direct runoff for a river basin. This model yields satisfactory result in prediction. Yaseen et al. (2016) in his work introduced a new data-driven model for streamflow forecasting, known as ELM. It was contrasted with other data-driven models like SVR and GRNN and observed to be significantly more superior to them with RMSE value around 21.3% less than SVR and roughly 44.7% less compared to GRNN. Li and Cheng (2014) combined the ELM with WNN for better monthly water discharge estimation in the river. They compared it with SLFN-ELM and SVM and discovered that SLFN-ELM performs slightly better in the prediction of the peak discharge and the taken WNN-ELM model yields more précised estimation compared to the other two models. Gupta et al. (2020) applied two asymmetric Huber loss function-based ELM model to deal with the noisy nature of the river SSL data. Experimental results expose that the ELM-based models were able to deal with the SSL datasets with high accuracy. Sadeghpour et al. (2014) proposed a hybrid model called a wavelet SVM (WSVM), which was a conjunction of

wavelet and SVM. It was found that WSVM could be used further as a prediction model for successful SSL prediction. Yadav et al. (2018) tried to forecast the SSL of Mahanadi River, India using a hybrid genetic algorithm-based artificial intelligence (GA-AI) model. In the comparison of this model with conventional models like MLR and SRC, it was found that the proposed GA-AI model yields better performance. Daneshvar and Bagherzadeh (2012) evaluated sediment yield using pacific southwest interagency committee (PSIAC) model and modified pacific southwest interagency committee (MPSIAC) model with the help of geographic information system (GIS) in Toroq watershed of Iran. Both models provided comparative outcomes and showed correlation coefficients with moderate level to the high level ($R^2 = 0.436$–0.996 to 0.893–0.998) for PSIAC as well as MPSIAC models, respectively. Rejaie-balf et al. (2017) applied a new parametric method called multivariate adaptive regression splines (MARS). It gave comparatively better performance compared to ANN, ANFIS, SVM, and M5 tree models. Choubin et al. (2018) used the CART model for modeling the SSL in a river. This model was compared with four common models: ANFIS, MLP neural network, radial basis function-SVM (RBF SVM), and proximal SVM (P-SVM). To evaluate the model capacities, various performance evaluation methods were used. As per the researcher, the CART model displayed the best results in estimating SSL, followed by RBF SVM. Kisi and Yassen (2019) implemented three ANFIS-based model to prove their usability in SSL estimation. Tarar et al. (2018) applied the Mann–Kendall test along with wavelet transform for SSL estimation in the upper Indus River and results show a very good R^2 value of 0.9. Gupta et al. (2018) tried to implement the KINEROS 2 model for forecasting streamflow and sediment load which yielded an average result. Very recent literature on SSL prediction using ANN includes Khan et al. (2019a, b), Nivesh et al. (2019), Yadav et al. (2020), Hazarika et al. (2020b), etc.

Predicting SSL through the GEP and ELM are some new techniques of artificial intelligence which had shown better performance over existing FFNN-BP technique. Notwithstanding when it is not feasible to create the mathematical function for the issue taken with the accessible soft computing methods, GEP could model it and thus wind up favorable during these circumstances over existing strategies. Another type of model known as SWAT was also implemented for calculating mean annual sediment precipitation. It showed an average result in SSL prediction (Oeurng et al., 2011). Morgan et al. (1998) applied a new model named the European soil erosion model (EUROSEM) for SSL estimation. However, it has a disadvantage that it is possible to be actualized only in smoothly incline railless planes, rilled surfaces, and crinkled surfaces. It was found by the researchers that EUROSEM overestimated the suspended sediment load

concentration, but the dissimilarity was not large. Tabatabaei et al. (2019) proposed a non-dominated sorting algorithm for SSL prediction from the dataset of Ramian hydrometric station on Ghorichay River. The results obtained from various SRC models suggest that the sediment rating curve-genetic algorithm-II model using non-dominated sorting algorithm-II gives better efficacy than the other models. Nourani et al. (2019) in their work proposed a wavelet-based data mining approach called a wavelet-M5 model for predicting the SSL of two different rivers named Lighvanchai and Upper Rio Grande. The obtained results in the Upper Rio Grande river reveal that the proposed wavelet-M5 model showed better performance compared to ANN, M5, and Nash Sutcliffe efficiency. Sharghi et al. (2019) suggested a novel wavelet exponential smoothing algorithm for estimating the SSL in the Lighvanchai and Upper Rio Grande rivers. Experimental results reveal that combining wavelet transform with exponential smoothing algorithm yields more precise results compared to WANN, ARIMA, and seasonal ARIMA models. Samet et al. (2019) tried to compare the performance among ANN, ANFIS, and GA, and noticed that among these models, the ANFIS showed the least error while predicting the SSL. Sharghi et al. (2019) suggested a hybrid emotional ANN (EANN) and wavelet transform conjunction model called wavelet EANN (WEANN) for river SSL prediction. The obtained results suggest that the model gives a good performance in estimating the SSL of Lighvanchai and Upper Rio Grande rivers.

The main intent of this paper is to present a brief discussion of the different artificial intelligence (AI)-based model that has been successfully applied for sediment load prediction. However, the main focus in on the studies between January 2015 and November 2020. Furthermore, to reveal the quality works that have been published between January 2015 and November 2020, a list of SCI/SCIE and Scopus indexed publication is also presented.

The rest of the paper is organized as follows: "Existing AI-based SSL estimation models" focuses on the major artificial intelligence (AI)-based models that have been fruitfully implemented from January 2015 to November 2020. The papers are obtained using the two queries "sediment load prediction" and "suspended sediment load prediction" in Google Scholar. "Experimental analysis" shows experimental analysis on two different SSL datasets that are collected from two different rivers in India. The last section is the conclusion and the future projection of the work. The details of the work that has been performed for SSL predictions are shown in Table 1. To be more specific, we have shown only the works that are indexed only in SCI/SCIE and Scopus using the two queries "sediment load prediction" and "suspended sediment load prediction" in Google Scholar. However, we have omitted ResearchGate as the recent research suggests that ResearchGate cannot still

challenge Google Scholar to provide early citation indicators. Moreover, although ResearchGate, in theory, allows automated data collection, unlike Google Scholar (except for Publish or Perish), its current maximum crawling speed is a major practical limitation on its use for large-scale data gathering (Thelwall and Kosha 2017). Moreover, Table 2 elaborates describes the performance evaluators that have been used by the researchers.

Existing AI-based SSL estimation models

The ANN

The property of working of brain to learn is studied and checked if it can be applied to the machine learning and gave rise to a very strong learning model known as neural networks or ANNs. ANNs are distributed, adaptive, and generally non-linear in nature built from many different processing elements (PEs). Each PE receives connections from other PEs and/or itself. Interconnectivity defines the topology of the system. Signals flowing through the connections are scaled by adjustable parameters called weights. PEs add up all of these contributions and produce an output that is a non-linear function of the sum. Outputs of PEs are either system outputs or sent to the same or other PEs (Rojas 1996). The value of ANNs stems from their expressive power, their ability to approximate functions, starting with the famous "Universal Approximation Theorem" according to which ANNs with depth 2, depending on their activation function, can theoretically approximate any continuous function in a compact domain to any level of accuracy (Cybenko 1989; Funahashi 1989; Hornik et al. 1989; Debao 1993; Barron 1994). This is done by emulating a non-linear process without actual knowledge of the model (Sharma and Lie 2012) and is capable of auto-adjusting in case conditions change in a time-dependent way (Lodge and Yu 2014) and of handling same or similar patterns (Wang et al. 2004). ANN are computationally difficult to train. On the other hand, modern neural networks are trained efficiently using stochastic gradient descent, backpropagation (BP), conjugate gradient descent, radial basis function (RBF), cascade correlation algorithm, etc., and a variety of tricks, including various activation functions (Livni et al. 2014). Goodfellow et al. (2015) have shown, for seven different ANN models of practical interest, that there is a straight path from initialization to solution that reduces objective function smoothly and monotonically. Recently, Bastani et al. (2016), Zhang et al. (2018), and Mangal et al. (2019) proposed new matrices for measuring the robustness of ANN. The ANN network's robustness is explicitly discussed in Bastani et al. (2016), Zhang et al. (2018), and Mangal et al. (2019). ANNs can

Table 1 List of contributions in SCI/SCIE and Scopus indexed journals from January 2015 to November 2020 using the queries "sediment load prediction" and "suspended sediment load prediction"

SI no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
1.	Himanshu et al. (2017b)	SVM with wavelet analysis	SVM	Marol and Muneru watershed	Asia	Daily	NMSE, R^2, NSE
2.	Özger and Kabatas (2015)	Fuzzy logic wavelet	Fuzzy logic	Corukhi river and East black sea basins	Asia and Europe	Monthly	NSSS, MAE, R
3.	Ulke et al. (2017)	GA_Brooks	Brooks, ANN, ANFIS	Major Rivers of the Aegean Region	Asia and Europe		MAE, MARE
4.	Hasanpour et al. (2019)	Fuzzy C means clustering-SVR (FCM-SVR)	SRC, ANN, ANFIS, SVR	Sistan river	Asia	Daily	RMSE, MAE, R^2, NSE
5.	Ramezani et al. (2015)	FF social-based algorithm	FF-independent component analysis, FF-GA, FF-particle swarm optimization	Maroon river	North America	Monthly	MSE, RMSE, MAE
6.	Shamaei and Kaedi (2016)	Linear genetic programming, NF	ANN	Valenciano, Quebrada Blanca	North America	Daily	RMSE, R^2
7.	Moeeni and Bonakdari (2018)	ARMAX-ANN	ARMAX, ANN	Cumberland river	North America	Daily	SI, MAE, MARE, CRM, VAF, R^2, AIC
8.	Olyaie et al. (2015)	Wavelet ANN	ANN, ANFIS, SRC	Flathead river, Santa Clara river	North America	Daily	R, CE, RMSE, MAPE
9.	Nourani and Andalib (2015a, b)	Wavelet LSSVM	WANN, ANN, LSSVM	Mississippi river	North America	Monthly	DC
10.	Nourani et al. (2016)	Two stage SVM	SVM	Ajichay river	Asia	Monthly	DC, RMSE
11.	Kumar et al. (2015)	ANN	–	Tropical Rainfall Measuring Mission (TRMM-3B42) dataset	Asia	Daily	CORR, RSR, E_{NS}
12.	Bouzeria et al. (2017)	ANN	–	Seybouse River	Africa	Daily	RMSE, MAE, R^2, NSE
13.	Chen and Chau (2016)	Hybrid double feedforward neural network	Multi-layer feedforward neural network (MFNN), double parallel feedforward neural network (DPFNN) and hybrid feedforward neural network (HFNN) models	Muddy Creek (A tributary of Colorado River)	North America	Daily	RMSE, NSEC
14.	Adib and Mahmoodi (2017)	GA-ANN	ANN	Marun river	Asia	Yearly	NMSE, R
15.	Tabatabaei et al. (2019)	Non-dominated sorting GA II	SRC	Ghorichay River	Asia	Daily	RMSE, MAE, NSE, R^2

Table 1 (continued)

Sl no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
16.	Himanshu et al. (2017a)	WA-SVM	SVM	Krishna river	Asia	Daily	RMSE, NMSE, NS, R^2
17.	Bayukildiz and Kameu (2017)	SVM, ANN, ANFIS	–	Coruh river	Asia	Daily	RMSE, MAE, EC, R^2
18.	Rashidi et al. (2016)	Gamma test-SVM	SVM	Korkosar basin	Asia	Daily	RMSE, NSEC, R^2
19.	Kisi and Kirmani (2016)	Adaptive neuro-fuzzy embedded fuzzy c-means clustering (ANFIS-FCM)	ANN, SRC	US Geological Survey (USGS) data	North America	Daily	RMSE, CE
20.	Samet et al. (2019)	ANN, ANFIS, GA	–	Gizlarchay River	Asia	Daily	PE
21.	Shiau and Chen (2015)	Quantile regression	–	Laonung station	Asia	Daily and yearly	MAE
21.	Alizadeh et al. (2017)	Multiple wavelet ANN	Single wavelet ANN	Toutle river	North America	Daily	RMSE, R^2
22.	Khan et al. (2019a, b)	FFBP, RBF, GRNN	–	Collected from "Indian Meteorological Department"	Asia	Monthly	MSE
23.	Bisoyi et al. (2019)	BPNN	–	Narmada river	Asia	Daily	MAE, SMAPE
24.	Sharghi et al. (2019)	Emotional-ANN (EANN) and wavelet- EANN	ANN	Upper Rio Grande and Lighvanchai Rivers	North America	Daily and monthly	NSSE, RMSE
25.	Rahgoshay et al. (2018)	SVM-GA	Multivariate adaptive regression spline (MARS) and MT tree model (M5T)	Royan and Vey-nakeh dams	Asia	Daily	RMSE
26.	Kisi and Yaseen, (2019)	Evolutionary fuzzy	NF inference system coupled with subtractive clustering (ANFIS-SC), grid partition (ANFIS-GP), and FCM (ANFIS-FCM)	Eel river	North America	Daily	RMSE, MAE, NSE, WI
27.	Barzegari and Barzegari (2016)	Markov and ARIMA	ANN	Gorganroud river	Asia	Monthly	RMSE, NMSE
28.	Salih et al. (2019)	M5P, attribute selected classifier (AS M5P), M5Rule (M5R), and K Star (KS)	–	Delaware River	North America	Daily	RMSE, MAE, NSE, PBIAS, R^2, RSR

Table 1 (continued)

SI no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
29.	Emamgholizadeh and Demneh (2019)	GEP, ANN, ANFIS	–	Kasilian and Telar rivers	Asia	Daily	MAE, R^2
30.	Ahmed et al. (2018)	Broyden–Fletcher–Goldfarb–Shanno (BFGS)	BP and local LR	Thal Canal	Asia	Weekly	RMSE, MSE, R^2, MBE
31.	Roushangar et al. (2016)	GEP and SVM	SRC	Housatonic river	North America	Daily	RMSE, NS, R^2
32.	Khan et al. (2019a, b)	ANN	–	Ramganga river	Asia	Daily	R^2
33.	Adnan et al. (2019)	Dynamic evolving neural-fuzzy inference system (DENFIS)	Adaptive neural-fuzzy inference system with fuzzy c-means clustering (ANFIS-FCM) and multivariate adaptive regression splines (MARS)	Jialing river	Asia	Daily	RMSE, MAE, NSE
34.	Tao et al. (2019)	Radial basis M5 model tree (RM5Tree)	Response surface method (RSM), artificial neural network (ANN) and M5Tree	Delaware river	North America	Daily	RMSE, R^2
35.	Tfwala and Wang (2016)	ANN	SRC	Shiwen river	Asia	Daily	R^2
36.	Malik et al. (2017)	CANFIS, MLP, MLR, MNLR and SRC	–	Pranhita river	Asia	Daily	RMSE, R, CE
37.	Sari et al. (2017)	ANN	–	Taboão river	Europe	Monthly	NS
38.	Malik et al. (2019)	Radial basis neural network (RBNN), self-organizing map neural network (SOMNN), least square support vector regression (LSSVR), and multivariate adaptive regression spline (MARS)	SRC	Godavari river	Asia	Daily	RMSE, COE, PCC, WI, PARE
39.	Bezak et al. (2017)	Copula function	MLR and exponential tree	Mura river	Europe	Daily	RMSE, MAE, NSE, R^2
40.	Choubin et al. (2018)	CART	SVM, ANFIS, ANN	Haraz river	Asia	Monthly	NSE

Table 1 (continued)

SI no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
41.	Mukhtar (2019)	Random forest, ANN, SVM	–	Tigris river	Asia	Daily and Monthly	RMSE, NSE, I^2
42.	Yilmaz et al. (2019)	ANN with artificial bee colony (ABC) and teaching–learning-based optimization (TLBO)	ANN, BP	Çoruh river	Asia and Europe	Daily	RMSE, MAE
43.	Meshram et al. (2019)	Two-phase feed-forward neuron network-particle swarm optimization gravitational search algorithm (FFNN-PSOGSA)	ANFIS, FFNN, FFNN-PSO	Narmada river	Asia	Monthly	MAE, NSE, WI
44.	Ebrahimi et al. (2015)	Honey-bees mating programming (HBMP), GEP	SRC	Mad river	North America	Daily	RMSE, MAE, R^2
45.	Gupta et al. (2020)	Asymmetric Huber loss function-based ELM (AHELM) and $\varepsilon-$ asymmetric Huber loss function-based ELM ($\varepsilon-$ AHELM)	SVR, ELM	Tawang Chu	Asia	Daily	RMSE, MAE
47.	Yadav et al. (2020)	Hybrid GA-based multi-objective optimization-ANN (GA-MOO-ANN)	ANN, MLR, SRC	Mahanadi river	Asia	Daily	MSE, RMSE, MAE, R, CE, ME, VAR
48.	Hazarika et al. (2020a, b, c)	Wavelet-twin SVR (TSVR) and Wavelet- ELM	ELM and TSVR (Peng 2010a; Peng 2010b)	Tawang Chu river	Asia	Daily	RMSE, MAE, SMAPE, MASE, SSE/SST
49.	Zounemat-Kermani et al. (2020)	GA-ANFIS, GA-SVR, ANFIS and SVR	SRC and MLR	Loíza river	South America	Daily	RMSE, MAE, NSE, R^2

Table 1 (continued)

Sl no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
50.	Ehteram et al. (2019)	Bat algorithm-ANFIS, weed algorithm-ANFIS, bat algorithm-multi-feedforward neural network, weed algorithm-multi-feedforward neural network	ANFIS and multi-feedforward neural network	Atrek river	Asia	Monthly	RSE, NSE, RSR, PBIAS
51.	Dutta and Sen (2018)	SWAT	–	Mahanadi river	Asia	Daily	R, RSR, NSE
52.	Liu et al. (2015)	SWAT	–	South Tobacco Creek (STC) watershed	North America	Daily and hourly	RMSE, R^2
53.	Kisi (2016)	EF	ANN, ANFIS-FCM	Eel river	North America	Daily	RMSE, MAE, R^2
54.	Martínez-Salvador and Conesa-García (2020)	SWAT	–	Upper Argos River	Europe	Monthly, Yearly	NSE, RSR, PBIAS
55.	Noori and Kalin (2016)	SWAT-ANN	SWAT and SWAT-Calibration Uncertainty Procedure (SWAT-CUP)	29 nearby watersheds in and around the city of Atlanta	North America	Daily	NSE
56.	Partovian et al. (2016)	ANN and ANFIS	MLR and ARIMA	Minnesota river	North America	Daily	DC
57.	McCarney-Castle et al. (2017)	Bayesian inference model (SIAR)	–	Lawsons Fork Creek basin	North America	Daily	R^2
58.	Zhang et al. (2015)	Wavelet based multi-resolution cointegration method	–	Weihe river	Asia	Yearly	RE
59.	Fadee et al. (2020)	Butterfly optimization algorithm (BOA) and GA	ANFIS, ANN, MLR	Eagle Creek watershed	North America	Daily	RMSE, MAE, R^2, NSE
60.	Zhang and Yang (2020)	CEEMDAN-GRU	GRU, LSTM, SVM	Huayuankou hydrological station	Asia	Monthly	MAE, MAPE, MSE
61.	Hazarika et al. (2020a, b, c)	Coiflet-OBELM and coiflet LDMR	OBELM, TSVR, WTSVR and LDMR	Tawang Chu river	Asia	Daily	RMSE, MAE, SMAPE
62.	Nhu et al. (2020)	Random subspace (RS)	RF, SVM	Haraz river	Asia	Monthly	NSE
63.	Babanezhad et al. (2020)	ANFIS	ACO-FIS	Cumberland river	North America	Daily	R

Table 1 (continued)

SI no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
66.	Shiri et al, (2020)	Wavelet-base processed signals and MARS	SRC	Three rivers of the USA	North America	Daily	NSE, VAF, R^2
67.	Chinnasamy and Sood (2020)	SWAT	SWAT	Kaligandaki Basin	Asia	Yearly	R, R^2, NSE
68.	Ehteram et al. (2020)	Whale algorithm, bat algorithm, PSO, multi-objective optimization	ANN, ANFIS, SVM and GP	Goorganrood River	Asia	Daily	R^2
69.	Kaveh et al. (2020)	LSTM	Feedforward NN, ANFIS	Schuylkill River	North America	Daily	RMSE, MAE, R^2
70.	Nourani and Andalib (2015a, b)	WLSSVM	LSSVM and ANN	Aji-Chay river	Asia	Monthly	DC
71.	Mehr and Sorman (2018)	Linear GP	ANN	Chai and Absardeh river	Asia	Daily	CE, RMSE
72.	Hazarika et al. (2020a, b, c)	SVR	ANN	Tawang Chu	Asia	Daily	MSE, RMSE, MAE
73.	Ghose (2018)	Radial basis neural networks (RBNN)	–	Subarnarekha basin	Asia	Daily	MSE, CE
74.	Toriman et al. (2018)	MLR, MLP and RBF	–	Jenderan basin	Asia	Daily	RMSE, MAE, CE, R^2
75.	Jaiyeola and Adeyemo (2019)	GEP	SRC	Inanda Dam	Africa	Monthly	RMSE, R^2
76.	Samantaray et al. (2020)	BPNN and ANFIS	ANN	Mahanadi river	Asia	Daily	R^2
77.	Mohamed and Shah (2018)	ANFIS, ANN	SRC	Thames river	Europe	Daily	MAE, RMSE, NSE
78.	Üneş et al. (2020)	SVM	GRNN	Omaha Station	North America	Daily	R, MAE, RMSE
79.	Yadav and Satyanarayana (2019)	Multi-objective	ANN	Mahanadi river	Asia	Daily	RMSE, R,
80.	Mustafa (2016)	Multiquadric basis function neural network	–	Bidor river	Asia	Daily	RMSE, MAE, CE
81.	Nikoo et al. (2018)	Feed forward Imperialist Competitive Algorithm (FF-ICA)	GA, PSO	Karoon river	Asia	Monthly	MSE, RMSE, R^2, MARE

Table 1 (continued)

SI no	Authors name	Model name	Model/models for comparison	River/Watershed/Basin	Continent	Time scale	Performance indicators
82.	Hamaamin et al. (2019)	ANFIS and Bayesian regression	SWAT	Saginaw river	North America	Daily	RSR, RMSE, R^2, NSE, PBIAS
83.	Sharghi et al. (2019)	Wavelet exponential smoothing, WANN	Exponential Smoothing, ARIMA and seasonal ARIMA (SARIMA)	Lighvanchai and Upper Rio Grande Rivers	North America	Daily and Monthly	NSE, RMSE, MARE
84.	Sattari et al. (2016)	GEP and ANN	–	Saied Abad watershed	Asia	Daily	MAE, RMSE, R^2, NSE
85.	Zende and Nagarajan (2015)	SWAT	–	Yerala river	Asia	Daily	–
86.	Talebi et al. (2017)	Regression trees and model trees	ANN and SRC	Hyderabad drainage basin	Asia	Daily	RMSE, R, MAE, SI, DR
87.	Taşar et al. (2017)	ANN, M5T, SRC, MLR	–	American Geological Research Survey Data	North America	Daily	MSE, MAE
88.	Haque et al. (2016)	GeoWEPP and WEPP	–	Langat river	Asia	Monthly	R^2
89.	Heng and Suetsugi (2015)	SRC	–	Mekong River	Asia	Monthly	NSE, PBIAS, RSR
90.	Tachi et al. (2020)	Regressions based on means of discharge	–	Seybouse basin	Africa	Yearly, seasonal and periodical	R^2

Table 2 Description of the performance indicators used in Table 1

Sl No	Indicator	Formula
1.	MSE (mean square error)	$\text{MSE} = \frac{\sum_{i=1}^{N}(f_i - o_i)^2}{N}$
2.	RMSE (root mean square error)	$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2}$
3.	MAE (mean absolute error)	$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|(f_i - o_i)\right|$
4.	R or CORR (Correlation Coefficient)	$R = \frac{\sum_{i=1}^{N}(f_i - \bar{f}_i)(o_i - \bar{o}_i)}{\sum_{i=1}^{N}(f_i - \bar{f}_i)\sum_{i=1}^{N}(o_i - \bar{o}_i)}$
5.	R^2 (Correlation Coefficient2)	$R^2 = \left(\frac{\sum_{i=1}^{N}(f_i - \bar{f}_i)(o_i - \bar{o}_i)}{\sum_{i=1}^{N}(f_i - \bar{f}_i)\sum_{i=1}^{N}(o_i - \bar{o}_i)}\right)^2$
6.	VAR (Variance)	$\text{VAR} = \frac{\sum_{i=1}^{N}(e_i - \bar{e}_i)^2}{N}$
7.	CE or COE or NS or NSE or NSSE or DC or ENS (Nash and Sutcliffe 1970)	$1 - \frac{\sum_{i=1}^{N}(f_i - o_i)^2}{\sum_{i=1}^{N}(o_i - \bar{o}_i)^2}$
8.	NNSE(Normalized NSE)	$\text{NNSE} = \frac{1}{2 - \text{NSE}}$
8.	RSR (RMSE observations standard deviation ratio)	$\text{RSR} = \sqrt{\frac{\sum_{i=1}^{N}(f_i - o_i)^2}{\sum_{i=1}^{N}(o_i - \bar{o}_i)^2}}$
9.	PCC (Pearson's correlation coefficient)	$\text{PCC} = \frac{\sum_{i=1}^{N}(f_i - \bar{f}_i)(o_i - \bar{o}_i)}{\sqrt{\sum_{i=1}^{N}(f_i - \bar{f}_i)^2 \sum_{i=1}^{N}(o_i - \bar{o}_i)^2}}$
10.	WI (Wilmot's Index)	$\text{WI} = 1 - \frac{\sum_{i=1}^{N}(f_i - o_i)}{\sum_{i=1}^{N}(f_i - \bar{o}_i)\sum_{i=1}^{N}(o_i - \bar{o}_i)}$
11.	PBIAS (Percent bias)	$\text{PARE} = \sum_{i=1}^{N}\frac{(f_i - o_i)}{o_i} \times 100$
13.	ME (Mean error)	$\text{ME} = \frac{\sum_{i=1}^{N}(f_i - o_i)}{N}$
14.	MBE (Mean biased error)	$\text{MBE} = \frac{1}{N}\sum_{I=1}^{N}\left(\bar{f}_i - \bar{o}_i\right)$
15.	PARE (Pooled average relative error)	$\text{PARE} = \sum_{i=1}^{N}\frac{(f_i - o_i)}{o_i} \times 100$
16.	SMAPE (Symmetric mean absolute percentage error)	$\text{SMAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|f_i - o_i|}{f_i + o_i}$
17.	MASE (Mean absolute scaled error)	$\text{MASE} = \frac{\text{MAE}}{\text{MAE}_{\text{naivemodel}}}$
18.	SSE/SST (Sum of squared error/total sum of squares)	$\text{SSE/SST} = \frac{\frac{1}{N}\sum_{i=1}^{N}(o_i - \hat{o}_i)}{\frac{1}{N}\sum_{i=1}^{N}(o_i - \bar{o}_i)}$
19.	SI (Scatter Index)	$\text{SI} = \frac{\text{RMSE}}{\bar{o}}$
20.	DR (Discrepancy Ratio)	$\text{DR} = \sum_{i=1}^{N}\frac{f_i}{o_i}$
21.	RE (Relative Error)	$\text{RE} = \sum_{i=1}^{N}\left|(f_i - o_i)/f_i\right|\bigg/ N$

Table 2 (continued)

Sl No	Indicator	Formula
22.	MARE (Mean absolute relative error)	$\text{MARE} = \dfrac{\sum\limits_{i=1}^{N}(f_i - o_i)}{N}$
22.	AIC (Akaike's information criterion)	$\text{AIC} = -2(\log - \text{likelihood}) + 2K$
23.	CRM (coefficient of residual mass)	$\text{CRM} = \dfrac{\left(\sum\limits_{i=1}^{N} f_i - \sum\limits_{i=1}^{N} o_i\right)}{\sum\limits_{i=1}^{N} f_i}$
24.	VAF (variance accounted for)	$\text{VAF} = \left(1 - \text{var}(f_i - o_i) \big/ \text{var}(f_i)\right)$

easily become unstable in the presence of disturbances or unmodelled dynamics. A constrained stable background algorithm (CSBP) was proposed by Korkobi et al. (2008) to overcome this situation. Furthermore, Haber and Ruhetto (2017) developed new forward propagation techniques to overcome the numerical instabilities in vanishing gradient problems of deep neural networks. Few other structural learning (SL)-based ANN architectures are the Cascade-Correlation learning (Fahlman and Lebiere, 1989) and the SL via forgetting (SLF) (Ishikawa 1996).

In the field of pattern classification and pattern recognition, ANNs have been effectively implemented (Bishop 1995) and are progressively utilized as a part of the studies taken in hydrology (Aly and Peralta 1999; Dawson and Wilby 1998; Zhang and Stanley 1997; Behzad et al. 2009). The embodiment of numerical-water drive models in ANNs was done by Dibike et al.(1999) for the problem of flow forecasting with positive results. In watersheds, ANNs are widely applied for soil erosion problem and rainfall–runoff relationship (Zhu et al. 1994; Tokar and Johnson 1999). As the utilization of ANNs developed in hydrological resources, a review of its idea as well as implementations were done by ASCE (2000) and inferred that ANN's execution is on a par with already operational models. By Freiwan and Cigizoglu (2005), it was applied in monthly river flow forecasting. Flood frequency analysis (1998), estimation of sanitary flows (1998), hydraulic characteristic of severe contraction (1998), and classification of river basins (2000) are some applications of ANN in other fields (Karunanithi et al. 1994; Grubert 1995; Venkatesan et al. 2009). Nagy et al. (2002) trained the ANN through deduced stream data for estimating the SSL in rivers. To calculate the output SSC (suspended sediment concentration), a network was established. This network has input variables like the Reynolds number, stream width ratio, Froude number (F_r), mobility number, etc., which were applied to calculate the load concentration. The commonly used models were compared with the ANN model on the output data. For comparison, the information of observed total load concentration (TLC) and calculated TLC through the predictor was used by:

$$D_r = T_o \big/ T_c. \tag{1}$$

In (1)

T_o = observed TLC and.

T_c = calculated TLC through the predictor.

ANN showed much better results than the most frequently used models. The calculated discrepancy ratio for Engelundand Hansen (1967) approach (2.34) had shown much more variations between T_c and T_o, whereas, for ANN (1.04), it was much closer. To predict the transportation rates of sediment load, an ANN-based method was introduced by Sarangi et al. (2005), which itself a data-driven model. Field data collected from several studies and published ravines having a high varying nature were taken to build or train the ANN model. The precision of estimation was observed to be superior to the models which were regularly utilized like Engelund and Hansen (1967). An ANN model was applied by Raghuwanshi et al. (2006) in Nagwan watershed for estimating the sediment load and overflow. Linear regression models were likewise produced for the examination of performance with the ANN. Here, every day and week by week drainage and sediment load was taken for prediction. The training data for both the models were of 5 years and the testing data were for 2 years. It was noticed that the ANN models outperformed the traditional methods like linear regression models. The ANN models for SSL prediction were also developed based on climate factors such as temperature, average rainfall, flow discharge, and the intensity of rainfall as these factors play a vital role in sediment depositions. Another ANN-based model was introduced by Zhu et al. (2007) based on these climate factors to simulate the monthly behavior of sediment depositions in Longchuanjiang River in China. The ANN model successfully simulated the monthly behavior of sediment depositions in Longchuanjiang River with nearly accurate results when proper variables were considered with the consideration of correlation of these variables with the suspended sediment depositions of the previous month. The conventional methods of prediction such as Multi Linear Regression (MLR) were also matched with the ANN models. In Alp

and Cigizoglu's research (2007), both these models were contrasted with each other based on their performance criteria. They took a couple of ANN models in which the BP learning algorithm and RBF algorithm were considered. The hydro-meteorological variables such as rainfall and flow and their relation with the daily SSL were examined by utilizing these two techniques of ANN by training it through the hydro-meteorological variables and SSL data taken from a catchment called Juniata in the United States. The outcomes implied that the performance given by ANN was much more accurate than MLR. To forecast the daily suspended sediment concentration, SRC, MLR, and ANN models were used by Rajaee et al. (2009) at a couple of gauging stations. The day-by-day waterway discharge and SSL information taken from these two stations utilized as the testing set for ANN. The outcomes generated from ANN model showed better results in comparison to the other models and the hysteresis phenomenon could be also simulated (Shiri and Kisi 2011). The conjunction of ANN with different approaches to make the predictions more precise to the measured value have also been done in the last decade. Geomorphology-based ANN (GANN) was modeled by Zhang and Govindaraju (2003), using morphological parameters to estimate the flow path probabilities for the prediction of runoff in a watershed. To estimate the flow path probabilities, a geomorphologic instantaneous unit hydrograph (GIUH) was applied. This graph could be developed through the engaged morphological parameters. To assign the synaptic, i.e., connection weights the path probabilities were applied to the hidden and the output layer. Hence, the application of other models along with the ANN showed that GANN performed more rationally and realistically. Soft computing tools were also used with the ANN to get more accuracy. According to Baskar et al. (2003), FFNN-BP performed better with five hidden layers with the use of GIS tools and ANN. Sarangi and Bhattacharya (2005) generated an ANN and a regression model using watershed-scale geomorphologic parameters for predicting sediment loss. While using the geomorphological based ANN, they found that the (coefficient of determination) R^2 values lying between 0.78 and 0.93 and efficiency factor (E) values in between 0.71 and 0.76, on the other hand, utilizing geomorphological based regression the R^2 numbers of 0.39–0.54 and E values of 0.53–0.46, and hence, it is concluded that ANN model was better concerning performance compared to regression models. Gharde et al. (2015) performed sediment yield modeling using the ANN model. The comparison of the performance of ANN with linear regression is done and they discovered that ANN concludes better accuracy compared to linear regression. Adib and Mahmoodi (2017), in his work, tried to predict ANN genetic algorithm (GA) and Markov chain hybrid model at flood conditions. Using GA, the

various ANN parameters are optimized. The researchers found that the normalized mean square error (NMSE) can be deducted by GA to 80%, but it does not significantly increase R. The water discharge (Q) and the suspended sediment concentration (SSC) were taken and their relationship was modeled by Khan et al. (2019b), in Ramganga river using ANN for SSC calculation. They concluded that ANN algorithm is efficient to model the relation between Q and SSC of a river. Moeeni and Bonakdari (2018) for the first time applied autoregressive moving average with exogenous terms (ARMAX) in conjunction to ANN for sediment load prediction. The ARMAX-ANN conjunction model achieved better outcomes than each ANN and ARMAX model (Choubin et al. 2017).

The physics of ANN changes with its training data and as it is all carried through hidden layer hence is not known to the user. The definition of an optimal network architecture of ANN and the knowledge of the internal system conditions are rigid as the user is not aware of the working of the hidden layer and no defined physical principles are available due to non-linearity of the input data. Hence, researchers faced difficulty in determining the appropriate ANN structure; therefore, they used the trial-and-error methodology to find the unit quantity of neurons working in the hidden layers. These analyses were broad and an expansive number of trials must be done to get the correct number of units. Due to its time-consuming property with similar operations application the trial-and-error approach resulted in the need for the development of some new methodology. The hydrodynamics could be integrated into the ANN models, so that the disadvantages arose due to the trial-and-error approach could be avoided and the problem of selecting an optimum ANN structure could be solved.

ANN overview

ANN is not a new approach as its development began nearly in the 1940s by McCulloch and Pitts (1943), to imitate a brain's way of functioning. It could be said that an ANN is a parallel-distributed information processing system. The information could be any raw data or a trained data. Its performance characteristic resembles the neural network formation inside the human brain.

Working of an ANN could be given in the following points:

1. The information is processed, at many single nodes, or elements or units known as neurons.
2. Connection links are established between nodes, and through them, signals are passed.
3. These connection links have weights assigned to them.

4. Non-linear transformations are implemented by the nodes to the aggregate input to get the aggregate yield (Jalalkamali et al., 2011).

A neural system is portrayed by its design that tends to the example of the links between the elements or neurons, its procedure for picking activation function, and the affiliated weights (Fausett 1994). Neural networks could be categorized based on layers: single, two-layer, and multi-layer, as well as on the basis of data flow. In multi-layer, the information flows from one layer to other layers, i.e., input for next layer are obtained from previous layer output and weights assigned to the connecting links, there is no relation between nodes in the same layer; whereas in recurrent ANN, the information runs in both ways from the input to the output as well as from output to the input side using the node (Bhattacharya et al. 2007; Ajmera and Goyal 2012; Barua et al. 2010).

The non-linear processes are mapped due to the use of SF in the network. SF is a non-decreasing, monotonic function. The simplicity of this function is obvious due to its derivative result; hence, it is easy to use during the testing procedure of ANN. The network of these above-defined nodes forms an ANN.

Training algorithms of ANN

The BP At Harvard University, an algorithm was proposed by Werbos (1974) in his PhD thesis known as BP algorithm. However, it was popularized when Rumelhart et al. (1988) trained the hidden layer neurons for a complex non-linear mapping problem. To train the ANNs, BP is the most popular algorithm which was used by many researchers.

BP is an algorithm which minimizes the error function. There are two passes in this algorithm, viz., forward pass and backward pass. It comes in the category of gradient descent technique. Here, the initial step is the forward pass where the accessible diverse set of input patterns is given to the input layer and its output is passed forward through the neural network to the hidden layer or output layer. Hence, the outcome acquired from the output layer is compared with the target output in focus and error between both these outputs is calculated (Govindaraju 2000). Now in the second step, i.e., backward pass; this error propagated back to the network, passing through every node and the weighted connections are updated accordingly as per the given equation:

$$\Delta w_{pq}(m) = -\eta^{\ell} \frac{\partial E}{\partial w_{ij}} + \kappa^{\ell} \Delta w_{pq}(m-1), \tag{2}$$

where $\Delta w_{pq}(m)$ as well as $\Delta w_{pq}(m-1)$ is the accretion in the weights between the nodes i and j in m^{th} and $(m-1)^{th}$ pass.

η^{ℓ} and κ^{ℓ} are learning rate as well as momentum, respectively.

A learning rate helps in reducing the likelihood of being caught in the local minima for the training procedure, and the momentum factor can accelerate the training procedure (Sahoo and Ray 2006; Freiwan and Cigizoglu 2005; Agarwal et al. 2009).

Even after the use of the learning rate the training process could be caught in local minima. The calculation to obtain minimum error is a slow training procedure as the solution traverses a zigzag path. Hence, a need for another training algorithm arose which could alleviate these factors.

The RBF In the application of neural networks, Broomhead and Lowe (1988) introduced a new function called RBF that could be used for training then after some years, Leonard et al. (1992) introduced a new training method to train the ANN utilizing RBF instead of the sigmoid function. As in the nervous system, some neurons how the characteristic of locally tuned response bounded to small range input space. RBF's working principle is also derived from the same concept.

This RBF neural network architecture is the same as normally used three-layer network models. In this model, a hidden layer is present and performing non-linear transformations without adjusting parameters. This hidden layer contains a parameter vector called 'centre'. This center could be calculated in many ways, one of the simplest ways is to pick it randomly from the available training samples, or it could be determined through the k-means clustering method, i.e., selecting the center of the different group's as the center or it could be adjusted through error correction training by considering it as a network parameter. For every node exist in the hidden layer, the Euclidean separation amongst center and the input vector is estimated and this Euclidean distance is changed via a non-linear function which decides the yield of concealed layer hubs, which are inputs to the output layer. At the output layer, these inputs are combined linearly to determine the ANN output for the ANN. Of an RBF-ANN, the output z could be calculated using the equation:

$$z = f(v) = \sum_{i=1}^{n} w_i R_i(x) + w_0. \tag{3}$$

In Eq. (3) w_i = weights assigned to the connections between neurons of the hidden layer and the output layer, x = the input vector, w_0 = bias

$R_i : R^n \Rightarrow R$ is an RBF which could be given as:

$$R_i(x) = \varphi \| x - \varsigma_i \|.$$

As it could be seen that the function $\varphi(.)$ will have the highest value at origin and decrease very quickly as its parameter goes to infinity, and it is also a requirement that $\varphi(.)$ should

approach zero. Generally, the class of RBF is narrated by Gaussian function given as:

$$R_i = -\exp\left(-\sum_{i=1}^{n} \frac{\|x - \varsigma_i\|^2}{2\sigma_{ij}^2}\right), \tag{5}$$

where $\varsigma_i^T = \left[\varsigma_{i_1}, \varsigma_{i_2}, ..., \varsigma_{i_n}\right]$ vector denotes the midpoint of the hidden layer, σ_{ij} denotes the width needed for Gaussian function.

The main difference between BP and RBF is the function used to tackle the associated nonlinearity in the available problem. In error propagation, the fixed-function sigmoid is used to implement the non-linearity, whereas the RBF uses the training dataset to implement the non-linearity, where it tries to find all hidden layer basis functions by itself and then in linear fashion summing all of them at output layer to give output.

Other algorithms are also available such as the cascade correlation algorithm. However, due to the unavailability of their application to predict SSL, they are not discussed here.

Advantages of ANN

1. Ability to learn by themselves and produce the outputs that are not limited to the provided input.
2. Fault tolerance.

Disadvantages of ANN

1. Unexplained network behavior
2. Determination of appropriate network structure (Mijwel 2018).

The GRNN

GRNN is an ANN algorithm in which there is no requirement of the iterative training procedure and there is no problem of local minima as encountered in feedforward backpropagation (FFBP) (Yin et al. 2016). The physically implausible estimates are mainly not generated by GRNN. To model rainfall-runoff, Cigizoglu et al. (2004) used three neural networks out of which one was GRNN. To forecast

and estimate the intermittent flow, Cigizoglu et al. (2004) applied the GRNN again to model river sediment yield. They applied the GRNN and compared its performance with MLR as well as SRC and showed that GRNN performance is the best among the three. Adnan et al. (2019) applied a novel dynamic evolving neural-fuzzy inference system (DENFIS) and proved its applicability in SSL prediction.

The model

Specht (1990) proposed a general regression neural system which does not need any iterative preparing method as in the BP model. In this model, an arbitrary function is approximated amid the input vectors and output vectors and is specifically evaluated from the training information. There is leverage appeared by GRNN that the error in estimation approaches to zero with the expansion in training set size by incorporating some mild limitations on the function. GRNN indicates predictable behavior and is fundamentally utilized in estimation issues of continuous variables as ordinarily standard regression strategies are utilized. GRNN follows the standard statistical methods. These methods are normally called kernel regression methods. Given a preparation set and the independent value i, it assesses the estimation of dependent variable p which is most likely and diminishes the mean squared error. The GRNN calculates the joint probability density function of i and p for a given training set.

The regression of p on I could be expressed as

$$E[p|I] = \frac{\int_{-\infty}^{\infty} yf(I,p)dp}{\int_{-\infty}^{\infty} f(I,p)dp}, \tag{6}$$

where $f(I,p)$ denotes the known joint pdf of vector I and p; I denotes the vector random variable; and p denotes the sample random variable.

In case, when density function $f(I,p)$ is not known, then through the observations samples of i and o, is estimated. A probability estimator $\hat{f}(I,p)$ could be computed based on sample values of i and p denoted by I^x and P^x, respectively. It could be given as:

$$\hat{f}(I,p) = \frac{1}{(2\pi)^{(q+1)/2}\sigma^{(q+1)}} \frac{1}{N} \times \sum_{i=1}^{n} \exp\left[-\frac{(I-I^i)^T(I-I^x)}{2\sigma^2}\right] \exp\left[-\frac{(P-P^x)^2}{2\sigma^2}\right]$$

$$\sum_{i=1}^{n} \exp\left[-\frac{(I-I^i)^T(I-I^x)}{2\sigma^2}\right] \exp\left[-\frac{(P-P^x)^2}{2\sigma^2}\right]. \tag{7}$$

In (16) q represents the dimension of the random vector variable i;

N represents the number of inspections for samples.

Each sample I^x and P^x have the sample probability of width σ which is assigned by the probability estimator $\hat{f}(I, p)$. The estimate for probability could be calculated as the aggregate of these probabilities (Specht 1990).

A scalar function Z_i^2 could be written as:

$$Z_i^2 = (I - I^x)^T (I - I^x). \tag{8}$$

Hence, substituting the values of Z_i^2 and performing the given integration, it yields the following expression:

$$\hat{P}(X) = \frac{\sum_{i=1}^{n} P^i \exp\left(-\frac{Z_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{Z_i^2}{2\sigma^2}\right)}. \tag{9}$$

This equation could directly be applied to the available arithmetic data. The initial layer of GRNN is the input layer where input quantities present. However, in the next layer, the pattern units or neuron elements are present which pass its outputs to the additional units in the summation layer. This summation layer is the third layer. The outputs of the summation layer are passed to the final layer, i.e., output layer. Here, output units calculate the final output for the GRNN (Kisi 2008).

Advantages of GRNN

1. Ability to handle noisy datasets.
2. Single-pass learning, no backpropagation required.

Disadvantages of GRNN

1. Big size.
2. Computationally complex (Mareček 2016).

Wavelet transform

The conjunction of wavelet analysis with the soft computing techniques had seen a rise in its use in the last decade. Numbers of studies were carried out by applying wavelet analysis and ANN in environmental engineering problems. The wavelet transform was developed nearly in the 1980s, but its utilization spread in recent years. To deal with nonlinear data, the existing conventional approaches were not as good as for linear data, and hence, the need for the conjunction of wavelet analysis with the traditional models arose. To predict droughts, Kim and Valdés (2003) introduced a wavelet ANN (WANN). Similarly, the conjunction of wavelet analysis with ANN in some other studies was studied by Tantanee et al. (2005) and Cannas et al. (2005) to predict annual rainfall and monthly rainfall–runoff, respectively in

Italy. WANN models and ANN models were also compared in different studies based on their performance in prediction. In the estimation of monthly streamflow, the two models WANN and ANN were compared by Cigizoglu and Kisi (2006) and concluded that WANN over performs the ANN. The ANN model performance was checked and evaluated with pre-processed data and without pre-processed data by continuous and discrete wavelet transforms again by Cannas et al. (2006), and it was concluded that ANNs with pre-processed data performed much more efficient way than the raw data. To estimate the SSL in waterways, Partal and Cigizoglu (2008) proposed a model with the conjunction of wavelets and neural networks. The un-decomposed raw data were measured and decomposed into wavelet components through the use of discrete wavelet transform (DWT), now on these wavelets components sum is performed selectively to result in a wavelet series. This wavelet series acted as an input vector for the ANN. It was shown that WANN predictions conveyed much more accurate results in comparison to traditionally used models, i.e., ANN and SRC. A model with the conjunction of wavelet and ANN was proposed by Nourani et al. (2009) to estimate the precipitation for 1 month ahead in Lingvanchai watershed situated in Tabriz, Iran. In this study, first, primary rainfall time-series was taken and the time-series was decomposed through the utilization of the wavelet analysis. After decomposition, the time-series for primary rainfall was converted into several multi-frequency time-series and this multi-frequency time-series was taken as the input vector to the ANN model. It was shown that the prediction of precipitation events may be for long term or short term can be done successfully because of the usage of several multi-frequency time-series as an input vector. Wavelet analysis was also combined with approaches like neuro-fuzzy (NF) and it was portrayed that it performed significantly better than the conventional approach, i.e., NF model. Rajaee (2010) predicted the daily SSL at a hydrological station of gauging located in the United States by applying a model in which wavelet conjunction with NF model was taken and known as Wavelet NF (WNF) model, in which the daily river discharge and time-series generated through suspended sediment was decomposed into numbers of time-series through the DWT function at different scales. Again, it was shown that WNF outperformed NF (Adamowski 2008; Rajaee 2011) combined wavelet with NF and found that WNF is an effective approach for river SSL prediction. Li and Cheng (2014) suggested a hybrid model which is the conjunction of ELM and WANN. They discovered that ELM gives better performance compared to SVM and the proposed WANN-ELM gives a more precise prediction compared to ELM and SVM.

A wavelet could be defined as a function in a mathematical form which is used to decompose the given continuous-time signal function into several scale components different

to each other where for every single scale component, a frequency range could be assigned. Each scale component will have different frequency range implying corresponding different resolutions; hence, each component could be studied with corresponding different resolutions. An oscillating waveform which is fast decaying and is of finite length known as mother wavelet. The mother wavelet is translated into multiple copies or scaled into different wavelets which are called daughter wavelets, and when a function is represented by wavelets, this is known as wavelet transform process. In the representation of functions that have discontinuities in their form and sharp peaks, the wavelet transforms show advantages over the traditionally used Fourier Transforms in the case of suspended sediment load prediction and reconstruction or deconstruction of the varying signals, nonperiodic, or of discrete nature. There are two types of wavelet transforms such as discrete wavelet transform (DWT) and continuous wavelet transform (CWT).

The CWT

It is a tool or an analytical formula used for dividing continuous-time signal or continuous-time function into daughter wavelets. Several wavelets can be reconstructed using the mother wavelet (MW). Let us consider $\chi(x)$ be the MW function which wavelet function can be obtained by the temporal translation τ and with dilation,d. The CWT of a continuous-time signal $x(s)$ may be expressed as (Ateeq- Ur-Rahman et al. 2018; Antoine 1998):

$$C(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(s)\chi^* \left(\frac{s - \tau}{a} \right) ds, \quad a > 0. \tag{10}$$

Here, * denotes the complex conjugate of $\chi(x)$ and $\chi(x)$ is the mother wavelet function. CWT seeks for correlation between the signal and wavelet.

To be classified as wavelet three criteria may be fulfilled by $\chi(x)$. They are:

1. $E = \int_{-\infty}^{\infty} |\chi(s)|^2 ds < \infty,$

where "| |" indicates the modulus operator that gives the magnitude of $\chi(x)$. If $\hat{\chi}(f)$ indicates the Fourier transform of $\chi(f)$, then the following condition must satisfy

2. $T_\psi = \int_{-\infty}^{\infty} \frac{|\chi(f)|^2}{f} df < \infty.$

T_ψ is the admissibility constant. The value of T_ψ depends on the chosen wavelet. To reconstruct the signal, the inverse CWT can be applied for the signal reconstruction as (Addison 2018; Zhang et al. 2020):

$$x(s) = \frac{1}{T_\psi} \int_{-\infty}^{\infty} \int_{0}^{\infty} C(a, \tau)\chi_{a,b}(s) \frac{da.d\tau}{a \times a}, \tag{11}$$

where $\tilde{\phi}(t)$ represents the dual function for $\varphi(t)$.

The DWT

In practical applications, the discrete-time signal is taken into account due to unavailability of the continuous-time signal. Here, the continuous-time signal is discretized with the use of the trapezoidal rule as mentioned above. If the data set of length is taken, then the DWT will produce coefficients. As the coefficients produced are square of the length of the taken dataset, it means that there is some redundant information present in the coefficients. Now, based on the problem, this redundant information could be utilized or may not be utilized. It is good to have redundant information, but sometimes it provides extra complexity. Occasionally, logarithmic uniform spacing (LUS) is used to tackle this redundant information problem. In this LUS, the resolution of β considered is coarser as compared to α scale discretization which results in N coefficients for length N. The DWT could be represented as:

$$\varphi_{r,s}(t) = \frac{1}{\sqrt{\alpha_0^r}} \varphi \left(\frac{t - n\beta_0 a_0^r}{\alpha_0^p} \right), \tag{12}$$

where r is an integer used to control the dilation in the wavelet, s is an integer used to control the translation in the wavelet, β_0 denotes the location parameter which takes its value always greater than 0

α_0 denotes a step finely dilated taking its value always greater than 1

Mainly, the values are taken in practice for α_0 and β_0 are 2 and 1 , respectively. For both the steps i.e., dilation and translation, if we take the power of two logarithmic scales, then it could be represented as:

$$\varphi_{r,s}(t) = 2^{-r/2} \varphi(2^{-r}t - s). \tag{13}$$

This is normally known as the 'dyadic grid' arrangement. Here, the above-mentioned equation for dyadic grid wavelet is taken in a compact form. Generally, the discrete dyadic wavelets are orthonormal to each other. There is no redundancy present in the signal which is regenerated from the wavelet transformed signal as the information stored in all the wavelet coefficients is not repeated. For a discrete-time-series,ω_i, the articulation for the dyadic wavelet change could be given as:

$$X(r, s) = 2^{-m/2} \sum_{i=0}^{N-1} \psi(2^{-m}i - s)\omega_i. \tag{14}$$

For the wavelet of discrete scale $\alpha = 2^r$, here, $X(r, s)$ represents the wavelet coefficient. In Eq. (14), x_i represents a finite time-series where $i = 0, 1, 2, ..., n - 1$ and n represents an integer power of 2, where $n = 2^m$. Hence, this displays the range for the variables r and s as β_0 and $1 < r < M$, respectively. It is enough and sufficient to use one wavelet to cover the time interval, and when the wavelet scale which is largest (i.e., 2^r where $r = m$) and creation of only one coefficient is needed. Hence, the same condition is applicable $r = 1$. At $r = 1_\alpha$ could take the value as 2^1, this infers that to convey the signal creation, 2^{M-1} or $n/2$ coefficients will occur at the same scale. It implies that if a discrete-time-series for above function having its length $n = 2^m$} is taken, and then, the summation of wavelet coefficients is given by $1 + 2 + 4 + 8 + ... + 2^{m-1} = n - 1$.

A component $\overline{X}$ remains the known smoothed component of the signal, which could be denoted by the mean of the signal. Hence, a time-series having its length $r = m$ is taken and is decomposed into $r = m$ components having no redundant information present in them.

The inverse discrete wavelet transform could be formulated as:

$$\omega_i = \overline{X} + \sum_{r=1}^{M} \sum_{s=0}^{2^{M-p}-1} X_{r,s} 2^{-r}/2 \psi(2^{-m} i - s). \quad (15)$$

Or simply, it could be formulated as:

$$\omega_i = \overline{X}(t) + \sum_{r=1}^{M} W_m(t), \quad (16)$$

where $\overline{X}(t)$ represents the approximate value of a sub-signal at any level m.

$W_m(t)$ denotes the coefficients for wavelet where $m = 1, 2, ..., M$.

These wavelet coefficients offer the details for sub-signals. Now, with these sub-signal details have the property to capture the small or it can be said that fine features in the values of data interpreted.

Here, $\overline{X}(t)$ is a residual term providing the information of the background for the available data. Due to the easiness quality of the $W_1(t), W_2(t), ..., W_m(t), \overline{X}(t)$, the number of properties can be easily considered using these components.

Advantages of WT

1. Shows simultaneous localization in time and frequency domain.
2. Fast computation while using fast WT.

Disadvantages of WT

1. Shift sensitivity.
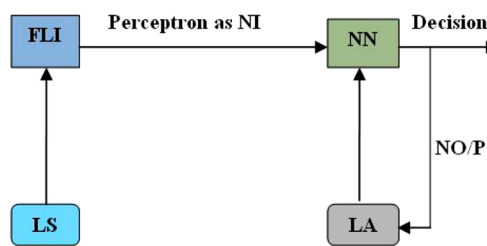2. Lack of phase information (Fernandes et al. 2003).

Fig. 1 The initial model of the fuzzy neural system (Fuller and Fullér 2000)

The NF

Neural networks perform excellently in recognizing patterns, but could not convey how these neural networks are reaching their decision. On the other hand, systems with the application of fuzzy logic efficiently explain the decisions taken by them, but do not have the property to automatically gain the rules to reach the decisions. Then, there are complex problems where there could be the presence of reasoning task as well as processing task which could be accomplished with fuzzy logic and neural network respectively. Therefore, it is better to use the hybrid model which could reason as well as the process in one single model, so that the complex problem could be solved with less effort. Hence, the need for a hybrid model such as NF approach has flourished which has the advantage of both the neural networks for processing as well as the fuzzy logic for decision-making and conveying.

There are numerous investigations performed to develop artificial intelligence techniques to simulate the problems available with inadequate physical knowledge of the systems. During the last decade, the use of fuzzy logic gained growth in the application of simulation problems like environmental uncertainties, river engineering, etc. As it is already mentioned, the application of ANN models in these non-linear problems shows its success widely. Still, we could not always rely only on one model; there is always a need for a different model about the chances of more accurate results. Hence, fuzzy logic (FL) is used to combine with these neural network learning algorithms in different estimation problems. This application of neural network learning algorithms on fuzzy modeling is normally known as NF modeling (Brown and Harris 1994). This model was implemented in many problems belonging to different areas like environmental engineering, financial trading, medical diagnosis, etc. Ocampo et al. (2007) applied a fuzzy model to model the ecological status in surface waters. Studies had been conducted to employ the neural network models with FL to arrive at a single hybrid model to evaluate the estimation of the SSLs. The fuzzy inference system (FIS) model is also applied in modeling the suspended sediments. The forecast of SSL was done by Tayfur et al. (2003) with the use of FL
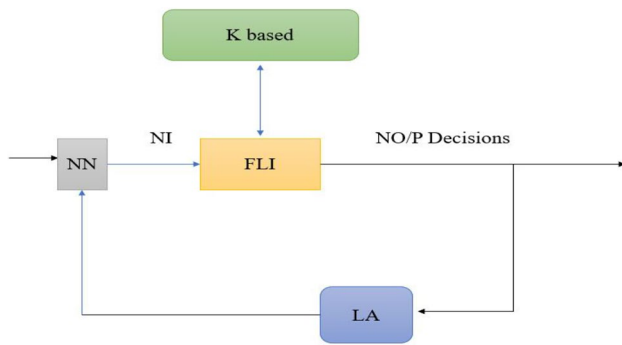
Fig. 2 The second model of the fuzzy neural system (Fuller and Fullér 2000)

on slope data and rainfall intensity from exposed soil surfaces. They concluded that the fuzzy approach provides better results over different slopes with various rainfall intensities and performed better for steep slopes. Lohani et al. (2007) compared the rating curve method with FIS for the performance in the simulation of a relationship for stage-discharge sediment concentration. The simulation was performed in a couple of gauging stations in a river called the Narmada in India. As expected, outcomes concluded that fuzzy method over performs the rating curve method. The accuracy in the estimation of monthly suspended sediments using different models was studied by Cigizoglu and Kisi (2006). The study was done in Salur, Koprusu, and Kuylus stations in Turkey. They compared ANN and SRC models with ANFIS for accuracy in estimation of suspended sediments, and the results exposed that NF System outperforms the other two models in estimation. Rajaee et al. (2009) compared MLR, ANN, NF, and SRC models for estimating the daily SSC. The examination was carried out in two hydrometer stations in USA. The data for sediment concentration and daily river discharge belonging to both stations had been implemented to train the models. The outputs showed that the NF model outperforms the other three models in predicting daily SSL.

Model

To model the fuzzy neural network in its computational process basically, these three steps are followed:

1. The fuzzy neural model is developed based on the working process of biological neurons.
2. The synaptic connections or the connection between neurons in each layer are modeled with fuzziness.
3. The adjustment of synaptic weights pertaining to the development of the needed learning algorithm.

There are two models which could be considered for NF modeling. In the first one, the fuzzy interface responds to the linguistic statements given and as output provides a quantity having direction as well as the magnitude to the multi-layer neural network, as shown in Fig. 1. Then this neural network (NN) tries to adapt itself to achieve the desired results through a learning algorithm. In the second model, first, the NN tune the membership functions which are used by the fuzzy system in the decision-making process, as shown in Fig. 2. The FL itself tune the membership functions directly using the required rules with linguistic statements, but it is computationally expensive. Hence, the performance could be improved with the use of neural network learning algorithms which would automate the tuning process.

In the above two figures.

FLI = fuzzy logic interface, NN = neural network, NI = neural input, NO/P = neural output, K based = knowledge-based, LA = learning algorithm, LS = linguistic statements.

The ANFIS and FL

Here, the model for adaptation of the second model (Fig. 2) is taken in a detailed manner which is also known by the name of ANFIS. This algorithm is an extraordinary instance of the second kind of modeling for NF-based models which were presented by Jang and Sun (1995). ANFIS follows the Sugeno-type fuzzy (SF) models. In this model, the reasoning mechanism attempts to determine the resultant function f for the provided input vector [i, j].

Here, an FIS having two inputs i, j and f as respective output is considered. In the initial order of the SF model, the knowledge used in the model has a form of if–then rules of FL, which can be shown as:

$$\text{Regulation 1 : If is } X1 \text{ and } j \text{ is } Y1, \text{ then} f_1 = l_1 i + m_1 j + r_1 \tag{17}$$

$$\text{Regulation 2 : If is } X2 \text{ and } j \text{ is } Y2, \text{ then} f_1 = l_2 i + m_2 j + r_2. \tag{18}$$

In (17) and (18), X_1, X_2 and Y_1, Y_2 are the membership functions for inputs i and j, respectively; l_1, m_1, n_1 as well as l_2, m_2, n_2 are the parameters of the resultant function (Firatand Gungor 2008).

The ANFIS functions are given as:

Layer I: In this node, outputs are defined hence the output OP_x^l could be given as

$$OP_x^l = \mu_{U_x}(i) \text{ for } x = 1, 2 \text{ or} \tag{19}$$

$$OP_x^l = \mu_{V_x-2}(j) \text{ for } x = 3, 4, \tag{20}$$

where i or j are the input nodes. U_x or $(V_x - 2)$ are the language statements or labels (high or low) which are

associated with the given node. These labels to the node are characterized as the membership functions from which it is true for any continuous and piecewise differential function, viz., triangular-shaped functions, Gaussian functions, generalized bell-shaped or normal distribution function, and trapezoidal-shape. Generally, the membership functions are given by normal distributed or bell-shaped functions for A and B. The output OP_i^l, at the node, could be calculated as:

$$OP_x^1 = \mu_{U_x}(i) = \frac{1}{1 + (i - {c_x}/{a_x})^{2b_x}}. \tag{21}$$

In (21), a_x, b_x, c_x is the set of parameters.

Layer II: Here, the incoming signal is multiplied at each node:

$$OP_x^2 = \mu_{U_x}(i)\mu_{V_x}(j), \quad x = 1, 2. \tag{22}$$

Layer III: Here, the normalized firing strength is computed for the ith node which could be expressed as:

$$OP_x^3 = \overline{w}_x = \frac{w_x}{w_1 + w_2}. \tag{23}$$

Layer IV: Here, for every node I, the benefaction of xth rule is computed toward the output of the model:

$$OP_x^4 = \sum_i \overline{w}_x f_x = \overline{w}_l(p_x i + q_x j + r_x). \tag{24}$$

In this equation, $\overline{w}_l$ is known as the output of layer III as well as $\{p_x, q_x, r_x\}$ is the collection of parameters.

Layer V: There is only 1 node present in the layer which computes the total outcome of the ANFIS model (Jang and Sun 1995; Nayak et al. 2004; Aqil et al. 2007) which could be shown as

$$OP_x^5 = \sum_x \overline{w}_x f_x = \frac{\sum_x w_x f_x}{\sum_x w_x}. \tag{25}$$

The learning algorithm used in the model is a hybrid algorithm in which two approaches, such as gradient descent and least squares, are encompassed and combined. This model takes a premise and consequent optimization parameter. In the first phase, the consequent parameter is established through node outputs in forwarding pass till the layer IV by the use of the least square approach. In the second phase, the errors are propagated backwards in the backward pass, and hence, through the use of gradient descent, the basic parameters are established accordingly (Jang and Sun 1995; Aqil et al. 2007; Zounemat-Kermani and Teshnehlab 2008).

Advantages of ANFIS

1. Compared to ANN, more transparent to the user.
2. Causes low memorization errors.

Disadvantages of ANFIS

1. Curse of dimensionality.
2. High computational cost.

The SVM

In recent times, an advanced approach in regards to computerized reasoning, known as SVM, has numerous implementations in learning strategy machines. This technique effectively has been utilized as a part of data arrangement and lately in regression issues. Cortes and Vapnik (1995) introduced SVM for problems related to binary classifications, and later, it has been applied in regression problems. Most of the studies on SVM tried to optimize the dual optimization problem, and it is very effective on both linear and non-linear datasets. Few SVMs show great results even if the data size is very large. This model was utilized for water management initially by Sivapragasam et al. (2001), Dibike and Solomatine (2001), and Zhao et al. (2002), and the novel model is known as SVM (Cristianini and Shawe-Taylor 2000; Chapelle 2007; Fung and Mangasarian 2003).

The SVR

SVR is also known as SVM for regression which is a regression method based on the support vectors, introduced by Vladimir Vapnik and his team in AT&T labs (Drucker et al. 1997). Mainly, SVR tries to minimize the generalization error using the SRM principle.

Suppose, the calibrating data $\{(i_1, j_1), \ldots (i_l, j_l)\} \subset \lambda \times \Re$, where λ denotes the input patterns count. The goal here lies in seeking a function $f(i)$ which has the highest ε deviation. This model is also known as ε-support vector regression:

$$f(i) = \langle w, i \rangle + z \text{ where } w \in \lambda, \ z \in \Re. \tag{26}$$

The primal problem of SVR may be stated as:

$$\min \frac{1}{2} \| w \|^2$$
$$\text{subject to} \begin{cases} o_l - \langle w, i_x \rangle - z \leq \in \\ \langle w, i_x \rangle + z - o_x \leq \in \end{cases}. \tag{27}$$

Sometimes errors are allowed, and therefore, slack variables ξ and ξ^* are introduced:

$$\min \frac{1}{2}\|w\|^2 + z\sum_{x=1}^{l}(\xi_x + \xi_x^*)$$

$$\text{subject to} \begin{cases} o_x - \langle w, i_x \rangle - z \leq \in +\xi_x \\ \langle w, i_x \rangle + z - o_x \leq \in +\xi_x^*. \\ \xi_x, \xi_x^* \geq 0 \end{cases} \tag{28}$$

The constant $z > 0$ determines the flatness trade-off between f and the maximum toleration of ε. This deals with $|\xi|_\in$ is known as ε- insensitive loss function (Noori et al. 2015):

$$|\xi|_\in = \begin{cases} 0 & if\,|\xi| \leq \in \\ |\xi| - \in & \text{otherwise} \end{cases}. \tag{29}$$

In practice, generally, the dual problem is solved rather than the primal problem. The dual formation can be inscribed as:

$$L_m = \frac{1}{2}\|w\|^2 + z\sum_{x=1}^{l}(\xi_x + \xi_x^*) - \sum_{x=1}^{l}(\eta_x \xi_x + \eta_x^* \xi_x^*)$$

$$- \sum_{x=1}^{l} d_x(\in +\xi_i - o_x + \langle w, i_x \rangle + z) \tag{30}$$

$$- \sum_{x=1}^{l} d_x^*(\in +\xi_x - o_x + \langle w, i_x \rangle + z)$$

subject to, $d_x^{(*)}, \eta_x^* \geq 0$

Partially deriving with reference to the Lagrangian variables (w, z, ξ_x, ξ_x^*) and substituting them in (39) give the dual optimization problem:

$$\max imize \begin{cases} -\frac{1}{2}\sum_{x=1}^{l}(d_x - d_x^*)(d_y - d_y^*)\langle i_x, i_y \rangle \\ - \in \sum_{x=1}^{l}(d_x + d_x^*) + \sum_{i=1}^{l} o_x(d_x - d_x^*) \end{cases} \tag{31}$$

$subject\ to\ \sum_{x=1}^{l}(d_x - d_x^*) = 0\ and\ 0 \leq d_x, d_x^* \leq C.$

Implementing the non-linear function using a kernel which is:

$$k(i_x, i_x^*) = \phi(i_x)^t \phi(i_x^*), \tag{32}$$

where $k(...)$ is a kernel function (Smolaand Schölkopf 2004). For any input space $\chi \in \Re$, its prediction is shown as:

$$f(i) = \sum_{x=1}^{l}(d_x - d_x^*)\,k(x_i, x_i^*) + z. \tag{33}$$
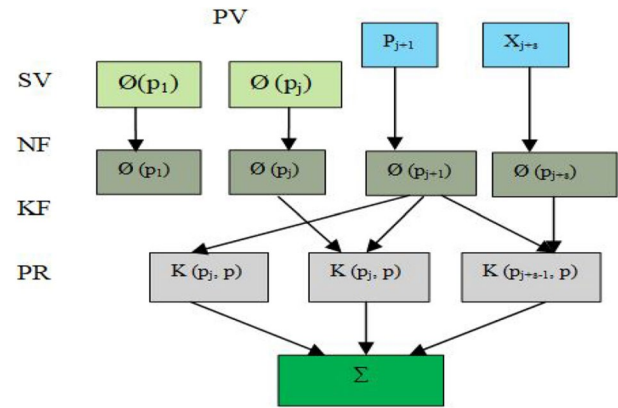
Advantages of SVM/SVR

Fig. 3 Architecture of the LSSVM model (Suykensand Vandewalle 1999; Nourani and Andalib 2015a)

1. High generalization ability.
2. It scales relatively well with high dimensional data.

   Disadvantages of SVM/SVR

1. Sensitive to noise and outliers.
2. High computational complexity (Hazarika and Gupta 2020).

The LSSVM

To take care of the non-linear classification and regression issues, SVM was updated and the new model was developed known as LSSVM. This model was first introduced by Suykens and Vandewale (1999) which has been vastly applied to the problems of work prediction and density prediction. The non-linear function of LSSVM could be written as:

$$f(p) = w^T \phi(\text{p}) + v, \tag{34}$$

where f is the association between the streamflow and SSL, w is called the weight vector with m dimension, and v is the bias factor (Nourani et al. 2017).

Due to the complicated nature of function error as well as fitting error, the regression issue might be offered by the basic minimization guideline as:

$$\min J(w,e) = \frac{1}{2}w^T w + \frac{\beta}{2}\sum_{j=1}^{m} e_j^2. \tag{35}$$

In (35), β represents the margin parameter.
The equation has the constraints:

$$Q_j = w^t \phi(P_j) + v + e_j (j = 1, 2, \dots, m). \tag{36}$$

In (36),e_j represents the negligible variable for P_j. This equation represents the optimization problem but with the constraints. Hence, to get the solution to the problem, these constraints can be converted into unconstrained problems in the objective function with the use of the Lagrange multipliers α_j as (Nourani and Andalib 2015a, b):

$$L(w,\ u, e,\ \alpha_j) = J(w,\ e) - \sum_{j=1}^{m} \alpha_j \{ w^T \phi(P_j)\ +\ v\ +\ e_j\ -\ Q_j \}.$$

(37)

ϕ denotes the mapping function. This function takes P and maps it into the m-dimensional feature vector. In Eq. (37), the partial derivatives could be taken with respect to w, u, e, and α_j, respectively, to reach the optimal conditions (Suykensand Vandewale 1999). It could be given as:

$$\begin{cases} w = \sum_{j=1}^{m} \alpha_j \phi(P_j) \\ \sum_{j}^{m} \alpha_j\ =\ 0 \\ \alpha_j\ =\ \beta e_j \\ w^T \phi(P_j)\ +\ v\ +\ e_j\ -\ Q_j \end{cases}.$$

(38)

Hence, the linear equations for (38) could be written as:

$$\begin{bmatrix} 0 & -Q^T \\ Q & ZZ^T\ +\ \frac{1}{\beta} \end{bmatrix} \begin{bmatrix} v \\ a \end{bmatrix}\ =\ \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

(39)

In (39)

$Q\ =\ Q_1,........,Q_{ym},$
$R\ =\ \phi(P_1)^T Q_1,...........,\phi(P_m)^T Q_m,$
$I = [1,.......,1],$
$\alpha = [\alpha_1,.......,\alpha_1].$

After using the kernel function,$K(P, P_j)\ =\ \phi(P_1)^T \phi(P_j),\ j\ =\ 1,..........,m,$ the LSSVM regressor becomes:

$$f(p) = \sum_{j=1}^{m} \alpha_j K(P,\ P_j)\ +\ v.$$

(40)

The RBF is generally utilized as a part of regression issues. The RBF kernel function is utilized as a part of the study as:

$$k(P, P_j) = e^{\left(-\frac{||P-P_j||^2}{2\sigma^2} \right)};$$

(41)

here, δ represents the parameter for RBF kernel. The estimation of this parameter is done through the network procedure itself. The universal architecture of LSSVM is illustrated in Fig. 3.

Here PV–Prediction vector, SV–Support vectors, KF–Kernel function, PR–Prediction results, NF–Non linear function.

Advantages of LSSVM

1. Good generalization performance.
2. Low computational cost.

Disadvantages of LSSVM

1. Sensitive to noise.
2. Sensitive to outliers.

The GA

Several new methodologies have been implemented to minimize the error rate in ANN, and eventually, they showed better performance. Among them, one of the most powerful methods is called the genetic algorithm. Although the algorithm consumes more time for training as compared to ANN, it achieves less erroneous results.

GAisthe types of computational models which are inspired by the functionality of genes. Though there are various applications of genetic algorithm, they are mainly viewed as function optimizers. GA provides different advantages to existing machine learning methods. For example, a GA.

i. Can be utilized by data mining for the field/attribute choice, and
ii. Can be attached with neural systems to decide ideal weights and design.

GA goes through three steps:

i. Build a population (typically chromosomes) of solutions and maintain it.
ii. Opt for better solutions for recombination among them.
iii. Use their offspring for replacing poorer solutions.

The general genetic algorithm operates as:

i. Initialization of a group of individual populations.
ii. Calculation of the fitness of each individual.
iii. Reproducing till a ceasing condition is not met.

Reproduction comprises of the following steps (Whitley 1994; Vankatesan et al. 2009):

i. Take at least one parent to reproduce.
ii. Make a mutation for selected individuals by making changes in a random bit of a string.
iii. Creating a new population.

Finally, one can conclude that the GA-based models are very effective for predicting the SSL.

Advantages of GA

1. Ability to avoid being trapped in a local optimum.
2. Use probabilistic selection rules rather than deterministic rules.

Disadvantages of GA

1. Computationally expensive
2. Low convergence (Aljahdali et al. 2010).

The GEP

GEP analogous to GA utilizes the individual population. Ferreira (2002) developed GEP which utilizes major standards of GA and genetic programming. Initially, it was developed for computer program generation. GEP is an evolutionary approach that emulates natural headway progress for influencing the PC platform to program stage and further to create a model (Baylar et al. 2011). The issues are encrypted in straight chromosomes of the same length as a PC program. GEP utilizes a large portion of the GA operators to perform the emblematic operation. However, some distinguishable dissimilarity can be noticed between GEP and GA. In GA, any numerical formula involves a symbolic representation of similar length (chromosomes) or components of non-linear nature. These components vary in their shapes and sizes, which are represented in the form of parse trees. Furthermore, this mathematical expression is encoded and represented in the form of expression trees (ET) in GEP. These expression trees consist of very simple fixed-length strings and are of various shapes and sizes. The encoding is done on these strings present in the mathematical expression (Ferreira and Gepsoft 2008; Cevik 2007). The algorithm of GEP starts by taking five segments. These segments are based on the arrangement of the functions, terminals, fitness function, controlling parameters, and stopping condition. In the following steps of the algorithm, a comparison is performed for estimated values and the original values. When the coveted outcome is accomplished, i.e., the taken criterion for the error is achieved, GEP stops. Few chromosomes are mutated to get new chromosomes by utilizing roulette wheel sampling if the desired error criterion could not be accomplished. The program stops and the chromosomes are decoded to get the best outcomes when the desired outputs are achieved (Teodorescu and Sherwood 2008).

Usually, the principal components in a GEP algorithm are the symbolically fixed-length strings of a mathematical formula known as chromosomes and the ET which carries relevant information. This information could be translated using conclusive language (e.g., Karva language) into expression trees which are the valuable features that permit to accurately deduce the genotype (Kayadelen 2011).

Gene comprises of two components, viz., head and tail. These components are mathematically expressed using some parameters alternatively known as variables, present in the head of gene. However, these parameters fall short of encoding mathematically, which give rise to the parameters used in the tail. The tail is present with required variables or constants to determine the difficulties to encode expressions as it is present with extra terminal symbols that help in encoding. The head usually consists of the arithmetic functions like addition (+), subtraction (−), multiplication (×), and division (÷), etc., while the tail consists of the independent variables or the constants like $(1, 2, 3, ..., a, b, c, x, y, ...)$. The length of the gene plays a vital role in the algorithm. Hence, it is decided at the starting of the analysis to define the total number of symbols present in both the head and tail. The ETs in the Karva language are read from left to right in a line and from top to bottom for whole of ET.

Advantages of GEP

1. Able to solve relatively complex problems using small population sizes.
2. Good generalization ability (Ferreira 2002).

Disadvantages of GEP

1. The conventional GA uses the method of fixed-length coding that performs poorly while facing complex problems (Cheng et al. 2018).
2. Low convergence.

The multiple regression (MLR and MNLR)

The MNLR

In MNLR, nonlinearity and multiple regression are the basic components for estimations of factual information. Linear regression (LR) in logarithmic space is generally used to decide the parameters of the derived equations:

$$\ln Z = \ln K + a_0 \ln I_0 + a_1 \ln I_1 + a_2 \ln I_2 + ... + a_n \ln I_n. \tag{42}$$

To make (42) non-linear in linear space, we can rewrite it as:

$$Z = KI_0^{a_0} X_1^{a_1} I_2^{a_2} ....I_n^{a_n}. \tag{43}$$

Equation (43) does not consist of intercept and various components, i.e., $I_0,...........I_n$(Tsykin 1984; Karim and Kennedy 1990). This method has been successfully implemented by a few researchers for SSL prediction.

The MLR

MLR models have influenced and controlled various fields for time-series estimation. MLR is generally utilized for modeling. For example, urban overflow toxin load, wash load silt concentrations, suspended sediment release, and the probability of swell capability of clayey soils. The main difference between MLR and simple LR (SLR) is that SLR has one predictor variables, whereas MLR has two or more predictor variables. In MLR, the dependent variables are dependent on p independent variables. These variables are often called explanatory variables. The equation for MLR could be given as:

$$y = \beta_0 x_0 + \beta_1 x_1 + ... + \beta_p x_p + \varepsilon. \tag{44}$$

In (44), $\beta_0, \beta_1, \beta_2, ..., \beta_p$ are the coefficients for the p independent variables representing the change in mean values (Rajaee et al. 2010; Toriman et al. 2018).

$x_0, x_1, x_2, ..., x_p$ represent the p explanatory variables or independent variables. y explains the variable to be predicted or the dependent variable. ε denotes the error. It follows the normal distribution with parameters $\mu = 0$ and σ^2.

The model fitting for MLR is considered with the addition of independent variables. The explained variance for the dependent variables will also increase when, i.e., R^2 increases.

Hence, the model may lead to over-fitting. Least square error criterion is the simplest choice to calculate the deviation between the desired value and the observed value. Hence, the model of MLR is said to be fit, only when the least square error is minimum. Different values of the coefficient β_i are taken to minimize the error.

This could also be represented in matrix form showing a more efficient structure of the model as there are a large number of predictor variables used in learning the model. Let us take a simple linear equation similar to Eq. (53), that is:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i. \tag{45}$$

For $i = 1, 2...........,n$, in (45), they could be written as:

$$y_1 = \beta_0 + \beta_1 x_1 + \varepsilon_1$$
$$y_2 = \beta_0 + \beta_1 x_2 + \varepsilon_2$$
$$.$$
$$.$$
$$y_n = \beta_0 + \beta_1 x_n + \varepsilon_n.$$

These equations could be written in matrix form as:

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ . & . \\ . & . \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ \varepsilon_n \end{bmatrix} \tag{46}$$

$$Y = X\beta + \varepsilon$$

Hence, n number of equations in (45) could be represented by just a simple Eq. (46), which is given above. The modeling of MLR can be used for prediction of SSL.

Advantages of multiple regression (MLR/MNLR)

1. Ability to determine the relative impact of one or more predictor variables on the value of the criterion.
2. Ability to identify outliers.

Disadvantages of multiple regression (MLR/MNLR)

1. Poor prediction performance (Maxwell 1975).
2. Sensitive to design anomalies in data (Akkaya and Tiku 2008).

The CART

Model

In the past, decision trees were proposed to work on the empirical examples to understand their performance on SSL prediction. However, this approach became popular with no strong theoretical foundations, because the CART model that is much more sophisticated and offers technical proofs for the results obtained. The merit of the CART model is that it could process both continuous as well as nominal attributes in both forms of the target and predictor variables as compared to other DT algorithms. In machine learning, data mining, and non-parametric statistics problems, CART outperformed the other traditionally used algorithms for classification. The CART is applied in many domains such as medical science, marketing research, river engineering,

and prediction problems. Besides, it is also applied in SSL prediction (Talebi et al. 2017).

The CART model applies a binary recursive partitioning procedure to the raw data. CART model was proposed by Breiman et al. (1984) to refer to both procedures, i.e., classification and regression. When the output to be predicted is a class, then it comes in classification category, and when the predicted output is any real number (like the price of a vehicle, age prediction), then it comes in the category of regression; it could be also said that if the predictor variable is of categorical form then CART gives classification and numerical form, then CART produces regression tree.

In this decision tree model, the tree initially grows without any stop to its maximum size and then pruning is performed split by split to the root, such that the model complexity could be minimized. The procedure of splitting and determining describes the procedure discrimination as classification and regression. In this model, as pruning is done split by split, hence the next split pruning will be the one which has the least complexity in tree performance for the available data for training. Trees produced will be invariant for any predictor attribute transformation. This model creates a grouping of nested trees. These all pruned trees are themselves candidate optimal trees. The calculation of predictive performance for each pruned tree is done and the tree with the best performance is taken as an honest tree. The tree selection is done based on independent test data depicting tree performance and not on any internal measurements. In case of unavailability of data or any cross-validation of data, the CART model would not give its fixed decision on the best tree selection. Instead, the CART model provides an automatic handle of missing values, balancing of class formation of dynamic features etc. (Breiman 2017). The split rule followed in CART is given by

If CONDITION TRUE

THEN go left

Else

go right

where the *CONDITION* could be represented as $X_i <= C$ and for a nominal attribute for continuous attributes and it expresses the membership in a definite set of values for a nominal attribute.

The CART mainly follows the Gini rule of impurity for classification over miss classification error and entropy index are included symmetrised costs if extended. It forms a set a randomly chosen element is arbitrarily labeled; following

the label distribution given in the subset the measure of Gini impurity tells how often this element is labeled mistakenly. If the target value is binary (i.e., 0/1), the Gini measure of impurity could be given as

$$g(t) = 1 - \backslash c(t)^2 - (1 - c(t))^2. \tag{47}$$

For class $1, c(t)$ represents the relative frequency inside the node in (47). And the gain produced due to the split of the parent node C could be given as

$$G(C) = g(C) - \alpha g(l) - (1 - \alpha)g(r). \tag{48}$$

In (48)
l and r represents the left and right children of C respectively.
α represents the fraction of instances which are going to the left children node (Timofeev 2004).

Two common impurity calculations are least squares and least absolute deviations for regression trees (Moisen, 2008).

Advantages of CART

1. Data normalization not required.
2. Intuitive.

Disadvantages of CART

1. High computational cost.
2. The small change of data can cause a large change in a tree structure.

The M5 Model Tree

M5 Model Tree type models were actualized in different hydrological implementations (Bhattacharya et al. 2007; Shrestha and Solomatine 2006). Quinlan (1992) states that the methods involving model trees represent the data in a structured form for a class and give the piecewise linear fit. Hence, generalizing the regression trees having constant leave values. Their structure is similar to the traditionally used decision tree structure. These model trees use linear regression functions at the leaves in place of discrete labels for different classes. This makes it perform well even for continuous several numbers of attributes. As it is normally done in the learning of decision tree models, M5 also learns similarly by dividing the available data in a tree-structured form based on the values of the predictive attributes. As the dimensions of the data set increase the computations requirement grows at a rapid rate. However, M5 could tackle the problems of a very huge amount of computations involving a large number of attributes. These are much smaller than the conventional regression trees which have less number of variables with clear decision strength (Frank et al. 1998; Singh et al. 2010; Goyal and Ojha 2011).

M5 model does not choose its attributes by the information-theoretic metric; instead, it tries to choose those attributes which could reduce the intra subset difference in the values. These values are the class values of the instances, and each branch of the tree goes downwards. When these values going downwards from the root to the leaf node, at each node, the attribute values of that node are tested for the expected reduction in the error. Furthermore, the value which maximizes this error is selected to calculate the standard deviation (SD) in the values. This SD is the measurement of the variability of the values. This splitting or division of data is stopped if the instance values reaching node have a very slight difference or the number of instances remained is very less in number (Goyal 2014; Goyal et al. 2013; Witten and Frank 2016).

The standard deviation reduction (SDR) could be evaluated as expressed below:

$$SDR = SD(T) - \sum_i \frac{T_i}{T} \times sd(T_x). \tag{49}$$

In (49) T represents the example set reaching the given node; T_x represents the example set delivering the output for the given set (Rejaie-balf et al. 2017).

Sometimes, there could be the formation of over-elaborating structures which needs to be pruned back due to unrelenting structures. It could be done by placing a leaf instead of a subtree. At these leaves which created after pruning, there could emerge some sharp discontinuities between neighboring straight models in a model where less number of training examples are utilized. Subsequently, the smoothening is performed in the conclusive stage. The update of adjacent conditions (linear) is performed, so that the outputs which are anticipated for the input vectors in correspondence to various conditions turn out to be about same in terms of value.

Advantages of CART

1. Can handle both numerical and categorical data.
2. Intuitive and easy to visualize.

Disadvantages of CART

1. Constrained to make only binary splits.
2. A small change in the dataset can make the tree structure unstable.

The CHAID

CHAID is a white box decision tree-based model that is used to search for the algorithms between a categorical response variable and another categorical predictor variable. This model was proposed by Kass (1980). This creates a decision tree using Chi-square statistics. It has the capability

of creating non-binary trees which implies that few splits achieve at least three branches, unlike the CART model. CHAID is successfully implemented in data mining, direct marketing, and medical diagnosis et cetera (Haughton and Oulabi 1997; Hill et al. 1997). Recently CHAID has been successfully implemented to predict the SSL along with SVM and ANN (Pektas and Dogan 2015). The obtained results revealed that the CHAID model was a better performer compared to SVM.

CHAID model proceeds stepwise:

i. First, the most favorable subset is taken for each predictor available in the decision tree.
ii. The second step follows with a comparison done between these predictors with their results, of which the best is taken.
iii. In the third step, the available data are further divided into subsets as per the chosen predictor.
iv. Finally, all these divided subsets are again analyzed without any dependency between them to get further subdivisions which are analyzed iteratively according to the above steps (Kass 1980).

Advantages of CHAID

1. Low computational cost.
2. Not constrained like CART to make binary splits.

Disadvantages of CHAID

1. To get reliable results, larger quantities of data are required.
2. Before analysis, real variables are forced into categorical bins (Nisbet et al. 2009).

The ELM

In machine learning, ANN is a great performer in terms of prediction as well as classification. It has been successfully applied in several fields that include hydrological forecasting. Compared to the traditional machine learning algorithm SLFN, ANN portrays satisfactory outcomes. However, ANN faces the problem of local minima and gives low generalization performance. The primary reason behind these issues is that moderate gradient descent algorithms are widely used and each parameter of the systems is tuned iteratively. Consequently, Huang et al. (2006) proposed another model called ELM to vanish these issues. Here, the weights and biases are arbitrarily relegated to the input layer and hidden layer separately. The output is estimated by utilizing the Moore Penrose generalized inverse of the hidden layer

output network. ELM gives better generalization performance and is fundamentally quicker than the ANN.

Suppose, a set is considered for training samples $\left\{ \left(i_x, o_x \right) \right\}_{x=1,....,m}$, and for each input example $i_x = \left(i_{x1}, o_{xn} \right)^t \in R^n$. Let $o_x \in \mathfrak{R}$ are the corresponding target values. For the arbitrarily allocated qualities of the learning parameters $a_s \in \left(a_{s1}, a_{s2}, ...a_n \right) \in R^n$ and $b_s \in R$ for the nodes present in the hidden layer, algorithm ELM computes its output function $f(.)$ as:

$$f(i_k) = \sum w_s E(a_s, b_s, i_k) = o_k \text{ for } k = 1, ..., m. \quad (50)$$

In (50),$E(a, b, i)$ represents the output function of the hidden layer. This output function is a piecewise continuous function showing the non-linearity in its nature and fulfills the states for all the inclusive estimate ability theorems.

$w = \left(w_1, ...w_l \right) \in \mathfrak{R}^n$ indicates the weight vector for the hidden layer connecting the nodes of the hidden layer to the output layer nodes. This vector is obscure in the knowledge of its working to the outer world, holding an ANN model property. The Eq. (50) can be modified to represent in matrix form as:

$$H = \begin{bmatrix} E\left(a_1, b_1, i_n \right) & ..... & E\left(a_n, b_n, i_n \right) \\ E\left(a_1, b_1, i_m \right) & ..... & E\left(a_n, b_n, i_m \right) \end{bmatrix}_{m \times n}. \quad (51)$$

For hidden layer,H represents the output matrix in the network as well as $o = \left(o_1, ...o_m \right)^t \in \mathfrak{R}^n$ gives the output vector values which are observed. Various activation functions can be used in ELM viz. sigmoid, multiquadric, ReLU, RBFetc.

If the output function, $E(a, b, i)$ of the hidden layer, is defined already in advance and assignment of values to the parameters $a_s \in \mathfrak{R}^n$; $b_s \in \mathfrak{R}^n$ is done randomly, then, to train the SLFN will be same as it happens in a rectangular system to obtain the solution for the least squares, i.e., $w \in \mathfrak{R}^l$. Here, this rectangular system will be linear. The generation of $w \in \mathfrak{R}^l$ is done explicitly as the solution for $b_s \in \mathfrak{R}^n$, in the form of least norm least squares. In this generated solution,$w \in H^+ y.H^+$ denotes the Moore–Penrose generalized matrix inverse of H(Balasundaramand Gupta 2014). Hence, the generated solution $w \in \mathfrak{R}^l$ will work as a fit model $f(.)$ for ELM regression. It could be expressed as

$$f(i_k) = \sum w_s E(a_s, b_s, i_k). \quad (52)$$

Advantages of ELM

1. Fast and efficient.
2. Parameter tuning is not needed.

Disadvantages of ELM

1. Noise and outlier sensitivity.
2. Overfitting problem.

The RF

RF is one of the most potent ensemble-based learning models. Breiman (2001) suggested the RF algorithm by adding additional randomness layer to the bagging method. It functions by constructing multiple decision trees and final predictions are extracted from the averaged results.

The algorithm of RF starts by drawing n_{tree} the bootstrap sample from the data. Afterward, an unpruned classification or regression tree is developed for each sample in the bootstrap (Ouedraogo et al. 2019). Subsequently, a random sample of the predictors is to be considered at each node and the best split from among those variables (predictors) is selected. Finally, new data are predicted by aggregating the prediction of n_{tree} trees (Liaw and Wiener, 2002).

Advantages of RF

1. Good generalization performance.
2. Can handle nonlinearity.

Disadvantages of RF

1. No interpretability.
2. Overfitting problem.

The MARS

The MARS model was formulated in the early 1990s by Jerome H. Friedman. The MARS system fits an adaptive non-linear regression model using multiple piecewise linear basis functions hierarchically ordered in consecutive splits over the predictor variable space (Spline 2013). The generalized form of the MARS model can be expressed as:

$$y = f(x) = c_0 + \sum_{n=1}^{N} c_N H_{kN}(x_{v(k,n)}), \quad (53)$$

where y is the output parameter, and c_o and N are the constant and the number of basis functions, respectively.

The basis function $H_{kN}(x_{v(k,n)})$ can be expressed as:

$$H_{kN}(x_{v(k,n)}) = \prod_{k=1}^{K} H_{kN}, \quad (54)$$

where $(x_{v(k,n)})$ is the predictor of the kth of the mth product.

Table 3 Statistics of the datasets (g/L)

Dataset	Mean	SD	Kurtosis	Skewness	Minimum	Maximum
Tawang Chu	0.07841	0.09939	6.92258	2.24453	0.005	0.6469
Pare	0.13451	0.07933	1.66322	1.01472	0.01736	0.41321

The ARMAX

The ARMAX model uses the linear input for prediction. The ARMAX model can be denoted as

$$ARMAX(\eta_a, \eta_e, \eta_m) = (1 - a_1 d^- ... {}^- a_{\eta_a} d^{\eta_a}) S_t = (1 - b_1 d^- ... - b_{\eta_e} d^{\eta_e})$$
$$+ Q_{t-k}(1 - c_1 d^- ... {}^- c_{\eta_m} d^{\eta_m}) \xi_t,$$

(55)

where S_t and Q_{t-k} are the predictor of the SSL and the discharge time-series, respectively. $(a_1, a_2 ..., a_{\eta_a})$ are the exogeneous coefficient vector. The ξ_t denotes the series of noise disturbance, $(c_1, c_2 ..., c_{\eta_a})$ is the moving average coefficient. $\eta_a, \eta_e,$ and η_m are the autoregressive, exogenous input, and moving average component, respectively. d and k are the predictor of the delay operator and the dead time in the system, respectively.

Advantages of ARMAX

1. Powerful model specially designed for time-series analysis.
2. Accurate and reliable forecast.

Disadvantages of ARMAX

1. Captures only linear relationships among variables.
2. Complex data pre-processing.

The fuzzy c-means clustering (FCM).

The FCM sections the dataset X into C clusters by minimizing the errors concerning the weighted distance of each data point x_i toward all centroids of the C clusters. Subsequently, the algorithm works as indicated by minimizing the objective function that is pigeonholed as:

$$\min J_{FCM} \sum_{c=1}^{C} \sum_{i=1}^{N} w_{ic}^e \| u_i - v_c \|$$

$$subject \ to \ \sum_{c=1}^{C} w_{ic}, \ i = 1, 2, ..., N,$$

(56)

where e represents the fuzzifier exponent $e > 1$, N is the total number of data points, w_{ic} represents the degree of belongings to the ith data point to the cth cluster which can be solved iteratively, v and u are the center of the cluster and the of data point that is provided as an input, respectively,

After initializing the center vectors, the centers can be recalculated until convergence as:

$$v_c = \frac{\sum_{j=1}^{N} w_{jc}^e u_j}{\sum_{j=1}^{N} w_{jc}^e} \ for \ c = 1, 2, ..., C \ and \ 1 < e < N.$$

(57)

Advantages of FCM

1. Fast convergence.
2. Gives the best result for the overlapped data set.

Disadvantages of FCM

1. Computationally expensive.
2. Sensitivity to noise and outliers.

The LDMR

The primal problem of LDMR (Rastogi et al. 2020) can be expressed as:

$$\min_{(w,b,\xi_1,\xi_2)} \frac{v}{2} \| Y - (K(G, G^t)w + eb) \|^2 + \frac{1}{2} Ce^t(\psi_1 + \psi_2) + \frac{d}{2} \| w \|^2,$$
$$s.t. Y - (K(G, G^t)w + eb) \le e\varepsilon + \xi_1, \ \xi_1 \ge 0,$$
$$(K(G, G^t)w + eb) - Y \le e\varepsilon + \xi_2, \xi_2 \ge 0,$$

(58)

where $\varepsilon, d, v > 0$ are the input parameters and $u = \begin{bmatrix} w \\ b \end{bmatrix}$;

$\| w \|^2 = u^t I_0 u$ where $I_0 = \begin{bmatrix} I & 0 \\ & \cdot \\ & \cdot \\ & \cdot \\ 0 & ...0 \end{bmatrix}$; $I^{m \times m}$ is an identity matrix;

$C > 0$ is the trade-off parameter; ψ_1 and ψ_2 are the slack variables. For obtaining the solutions from (58), Lagrange's multipliers are introduced as:

$\alpha_1 = (\alpha_{11}, \alpha_{12}, ..., \alpha_{1m})^t$ and $\alpha_1 = (\alpha_{21}, \alpha_{22}, ..., \alpha_{2m})^t$.

The dual formulation of (58) may be expressed as:

$$\min_{(\alpha_1,\alpha_2)} \frac{1}{2} (\alpha_1 - \alpha_2)^t Z_0 (dI_0 + vZ_0^t Z_0)^{-1} Z_0^t (\alpha_1 - \alpha_2)$$
$$+ y^t Z_0 (dI_0 + vZ_0^t Z_0)^{-1} Z_0^t (\alpha_1 - \alpha_2)$$
$$- Y^t(\alpha_1 - \alpha_2) + \varepsilon \ e^t(\alpha_1 + \alpha_2)$$
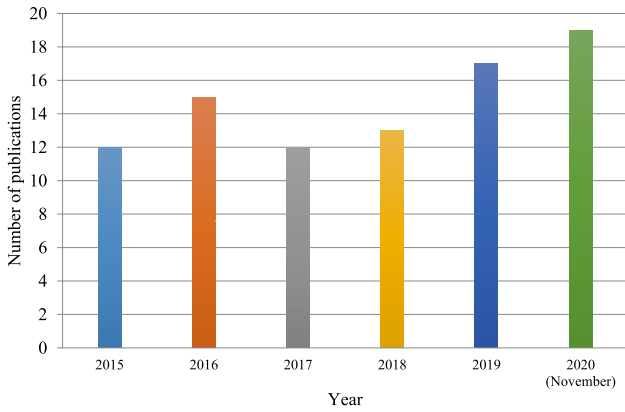$$s.t. 0 \le \alpha_1 \le Ce \ and \ 0 \le \alpha_2 \le Ce,$$

(59)

Fig. 4 Number of published journal papers regarding SSL prediction (indexed in SCI/SCIE and scopus) with respect to year of publication to best of our knowledge
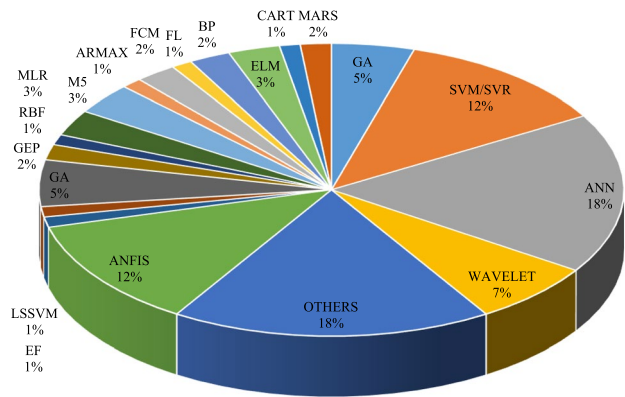

Fig. 5 Pie-chart showing various implemented models for SSL prediction from 2015 to 2020 (November) (indexed in SCI/SCIE and Scopus)
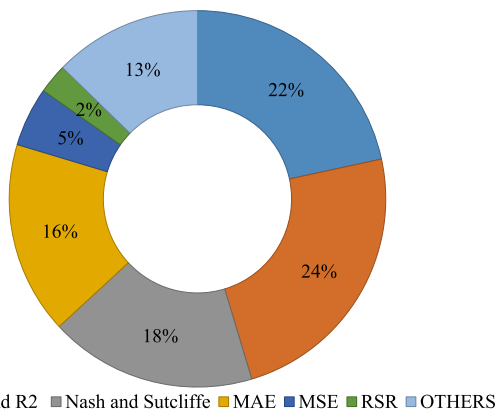

Fig. 6 Doughnut chart showing the applied percentage of the various performance evaluators for SSL prediction from 2015 to 2020 (November)

where, $Z_0 = [\,K(G, G^t)\ e\,]$ and be an augmented matrix (Hazarika et al. 2020b).

For a new instance x, the decision function $\varphi(.)$ is achieved as follows:

$$\varphi(x) = K(x^t, G^t)w + b.$$

Advantages of LDMR

1. Insensitive to noise and outliers.
2. Handles non-linearity.

Disadvantages of LDMR

1. Computationally expensive.

In Table 3
$N =$ Total samples

$f =$ predicted outcome

$\bar{f} =$ predicted outcome (mean)

$o =$ observed value

$\bar{o} =$ observed value (mean)

$\hat{o} =$ predicted value of o

$e =$ error

$\bar{e} =$ error (mean)

$K =$ the number of model parameters

$\log - \text{likelihood} =$ measure of model fit.
The higher the number, the better the fit.

The year-wise publications in SCI/SCIE and Scopus indexed journals are portrayed in Fig. 4 from January 2015 to 2020 (November). It is noticeable from Fig. 4 that there is an increase in the number of good publications from 2018. Figure 5 shows the Pie-Chart for various prediction models that have been applied during the time range from January 2015 to November 2020. Figure 6 exhibits the various performance measures that have been used for evaluating the model performances. One can observe from Fig. 6 that R and R^2 are the most widely accepted performance measure for model evaluation which is followed by the RMSE. However, all these representations are approximate.

Table 4 Obtained RMSE for a few models using the two SSL data from two different rivers

Dataset	SVR	LSSVR	TSVR	ELM	OB-ELM	LDMR
Tawang Chu	0.04013	0.11061	0.041603	0.04418	0.03952	0.04065
Pare	0.20677	0.24652	0.20829	0.20945	0.19926	0.20703

Table 5 Obtained MAE for a few models using the two SSL data from two different rivers

Dataset	SVR	LSSVR	TSVR	ELM	OB-ELM	LDMR
Tawang Chu	0.01464	0.05518	0.01749	0.04743	0.01608	0.01559
Pare	0.12338	0.18945	0.12983	0.14525	0.12237	0.1246

Experimental analysis

The experiments are performed on a desktop computer system on MATLAB 2019a software with 32 GB RAM, 3.20Ghz Intel i-7 processor on Windows 7 operating system. The QPP problems of SVR, TSVR, OB-ELM, and LDMR models are solved using the quadprog function in MATLAB. The datasets are randomly split, such that 70% are used for training and the remaining 30% used for testing. The prediction errors based on RMSE and MAE are revealed in Tables 4 and 5, respectively. The optimum values of the regularization parameter,C of the SVR, TSVR and ILTPISVR models are chosen from a range of parameters $\{10^{-5}, 10^{-4}, .., 10^5\}$. Moreover, for the LDMR and the proposed MKLDMR and MHKLDMR models, the optimal regularization parameters C, $C_1 = C_2$ are also chosen from $\{10^{-5}, 10^{-4}, .., 10^5\}$. For all of the models, the value of the ε parameter is chosen from a range of $\{0.05, 0.1, 0.5, 1, 1.5, 2\}$. For computational convenience, the k parameter of the LDMR is fixed to 1. The optimum values of L parameter is considered from $\{20, 40, 50, 100, 200, 500\}$. We have performed experiments using two SSL datasets that are collected from two different rivers in India, i.e., Pare river and Tawang Chu river. The Tawang Chu river dataset contains SSL data from January 1, 2015, to December 31, 2015, whereas the Pare river dataset contains SSL data from December 12, 2018, to 5 May 2019. The details of the datasets are expressed in Table 3.

Few conventional AI models such as SVR, TSVR, ELM, OB-ELM, and LDMR has been applied on the two different SSL datasets. The results obtained based on RMSE and MAE are shown in Tables 4 and 5, respectively. It can be noticed that the AI model shows good prediction performance for the datasets. The observed SSL versus predicted SSL plots are shown in Fig. 7 for the reported models.

It can be observed from Fig. 7 that the R^2 value is low for the reported models. To improve the prediction performance of the reported models, different types of decomposition methods such as variational mode decomposition (VMD) (Dragomiretskiy and Zosso 2013) and ensemble empirical mode decomposition algorithms (EEMD) (Wu and Huang 2009) are suggested as a data pre-processing step. Also embedding the DWT to the conventional AI-based models can improve the prediction performance of the same (Zhu et al. 2016; Hazarika et al. 2020a, b, c).

Summary and future projections

Sediment load prediction is one of the prime issues in hydrology. The study of SSL characteristic is a very cumbersome process due to its non-linear nature. Models like MLR and MNLR have been used to tackle the non-linearity of the problem and succeeded in a great way. However, these models could not give great accuracy in prediction. However, these models could be tried in conjunction with other learning models and needs to be checked for their prediction accuracy. The different models applied for prediction of SSL cannot be compared strictly based on their performance. Since the different field conditions may alter the performance of the same model with accuracy in some condition and average in some other condition, it could give average accuracy. Therefore, it could be inferred that models like ANN, SVM, GEP, GA, ANFIS, and their hybrid models like ANFIS-ANN, WANN, etc., have their specialities and could perform better with the prevailing conditions. It is also noticed the application of wavelet transforms on the input available giving rise to daughter wavelets for different time-series. These daughter wavelets are also feasible to use.

The principal task of researchers has been to somehow study the non-linear nature of sediment loads using a simple learning algorithm, so that the learning process could be understandable to the outer world. However, it creates a paradoxical situation in models like ANN. Hence, to get better accuracy with a simplified learning process some other algorithms are also applied, viz., ELM, GEP, and so on. Out of these algorithms, ELM as well as wavelet-based models have been showing great potential and could be used
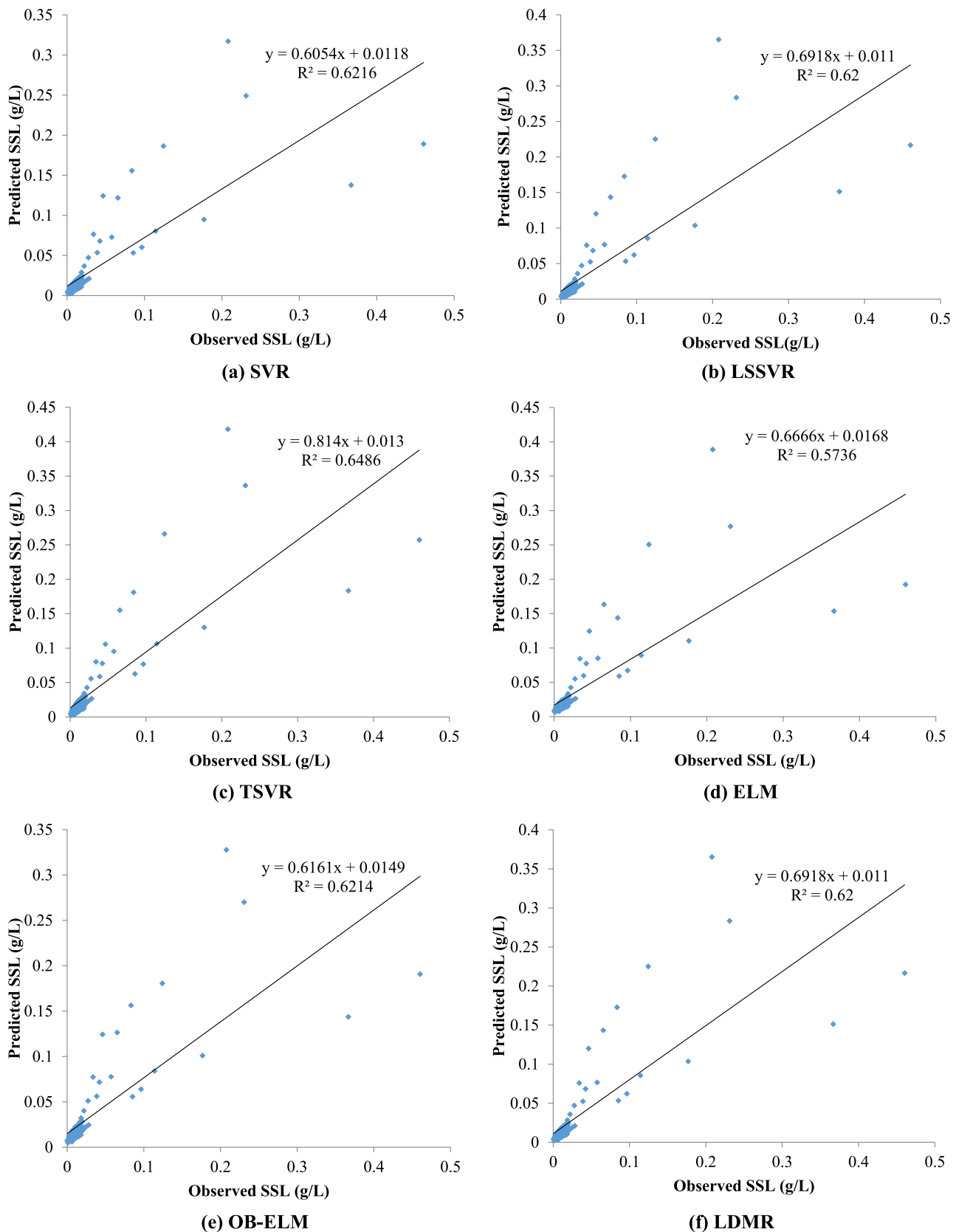
Fig. 7 Observed SSL (g/L) versus predicted SSL (g/L) plot of a few AI-based models on river SSL dataset collected from the Tawang Chu river. **a** SVR, **b** LSSVR, **c** TSVR, **d** ELM, **e** OB-ELM, and **f** LDMR

by integrating it with different learning models. Also, the effectiveness of the random vector functional link (RVFL) could be tested for SSL prediction. These hybrid models perhaps could give better accuracy and could be tackled much more efficiently using different learning algorithms.

Declarations

Conflict of interest Authors have no conflict of interest.

References

Adamowski JF (2008) River flow forecasting using wavelet and cross-wavelet transform models. Hydrol Process 22(25):4877–4891

Addison PS (2018) Introduction to redundancy rules: the continuous wavelet transform comes of age. Phil Trans R Soc A. https://doi.org/10.1098/rsta.2017.0258

Adib A, Mahmoodi A (2017) Prediction of suspended sediment load using ANN GA conjunction model with Markov chain approach at flood conditions. KSCE J Civ Eng 21(1):447–457

Adnan RM, Liang Z, El-Shafie A, Zounemat-Kermani M, Kisi O (2019) Prediction of Suspended Sediment Load Using Data-Driven Models. Water 11(10):2060

Agarwal A, Rai RK, Upadhyay A (2009) Forecasting of runoff and sediment yield using artificial neural networks. J Water Resour Prot 1(05):368

Ahmed F, Hassan M, Hashmi HN (2018) Developing nonlinear models for sediment load estimation in an irrigation canal. Acta Geophysica 66(6):1485–1494

Ajmera TK, Goyal MK (2012) Development of stage–discharge rating curve using model tree and neural networks: an application to Peachtree Creek in Atlanta. Expert Syst Appl 39(5):5702–5710

Akkaya AD, Tiku ML (2008) Robust estimation in multiple linear regression model with non-Gaussian noise. Automatica 44(2):407–417

Alizadeh MJ, Nodoushan EJ, Kalarestaghi N, Chau KW (2017) Toward multi-day-ahead forecasting of suspended sediment concentration using ensemble models. Environ Sci Pollut Res 24(36):28017–28025

Aljahdali SH, Ghiduk AS, El-Telbany M (2010) The limitations of genetic algorithms in software testing. In: ACS/IEEE International Conference on Computer Systems and Applications-AIC-CSA 2010, IEEE, pp 1–7

Al-Mukhtar M (2019) Random forest, support vector machine, and neural networks to modeling suspended sediment in Tigris River-Baghdad. Environ Monit Assess 191(11):673

Alonso CV, Neibling WH, Foster GR (1982) Estimating sediment transport capacity in watershed modeling. Trans ASCE 24(5):1211–1226

Alp M, Cigizoglu HK (2007) Suspended sediment load simulation by two artificial neural network methods using hydro-meteorological data. Environ Model Softw 22(1):2–13

Aly AH, Peralta RC (1999) Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. Water Resour Res 35(8):2523–2532

Antoine JP (1998) The continuous wavelet transform in image processing. CWI Q 11(4):323–345

Aqil M, Kita I, Yano A, Nishiyama S (2007) Analysis and prediction of flow from local source in a river basin using a Neuro-fuzzy modeling tool. J Environ Manag 85(1):215–223

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000) Artificial neural networks in hydrology. I: Preliminary concepts. J Hydrol Eng 5(2):115–123

Ateeq-Ur-Rehman S, Bui MD, Rutschmann P (2018) Variability and trend detection in the sediment load of the Upper Indus River. Water 10(1):16

Awad M, Khanna R (2015) Support vector regression. Efficient learning machines. Apress, Berkeley, pp 67–80

Azamathulla HM, Ghani AA, Chang CK, Hasan ZA, Zakaria NA (2010) Machine learning approach to predict sediment load–a case study. Clean: Soil, Air, Water 38(10):969–976

Babanezhad M, Behroyan I, Marjani A, Shirazian S (2020) Artificial intelligence simulation of suspended sediment load with different membership functions of ANFIS. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05458-6

Balasundaram S, Gupta D (2014) 1-Norm extreme learning machine for regression and multiclass classification using Newton method. Neurocomputing 128:4–14

Barron AR (1994) Approximation and estimation bounds for artificial neural networks. Mach Learn 14(1):115–133

Barua S, Perera BJC, Ng AWM, Tran D (2010) Drought forecasting using an aggregated drought index and artificial neural network. J Water Clim Change 1(3):193–206

Barzegari F, Barzegari F (2016) Suspended sediment prediction using time series and artificial neural networks models (case study: Ghazaghly Station in Gorganroud River). J Watershed Manag Res 6(12):216–225

Baskar RS (2003) Estimation of watershed runoff using artificial neural networks (Doctoral dissertation, Indian Agricultural Research Institute; New Delhi).

Bastani O, Ioannou Y, Lampropoulos L, Vytiniotis D, Nori A, Criminisi A (2016) Measuring neural net robustness with constraints. Advances in neural information processing systems. Springer, Berlin, pp 2613–2621

Baylar A, Unsal M, Ozkan F (2011) The effect of flow patterns and energy dissipation over stepped chutes on aeration efficiency. KSCE J Civil Eng 15(8):1329–1334

Behzad M, Asghari K, Coppola EA Jr (2009) Comparative study of SVMs and ANNs in aquifer water level prediction. J Comput Civ Eng 24(5):408–413

Bezak N, Rusjan S, KramarFijavž M, Mikoš M, Šraj M (2017) Estimation of suspended sediment loads using copula functions. Water 9(8):628

Bhattacharya B, Price RK, Solomatine DP (2007) Machine learning approach to modelling sediment transport. J Hydraul Eng 133(4):440–450

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Bisoyi N, Gupta H, Padhy NP, Chakrapani GJ (2019) Prediction of daily sediment discharge using a back propagation neural network training algorithm: a case study of the Narmada River India. Int J Sediment Res 34(2):125–135

Bouzeria H, Ghenim AN, Khanchoul K (2017) Using artificial neural network (ANN) for prediction of sediment loads, application to the Mellah catchment, northeast Algeria. J Water Land dev 33(IVVI):47–55

Breiman L (2001) Statistical Modeling: The two cultures (with comments and a rejoinder by the author). Stat Sci 16(3)

Breiman L (2017) Classification and regression trees. Routledge, London

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth Int. Group 37(15):237–251

Brooks NH (1965) Calculation of suspended load discharge from velocity and concentration parameters. Proc Federal Interag Sediment Conf Misc Publ 970:229–237

Broomhead DS, Lowe D (1988) Radial basis functions, multi-variable functional interpolation and adaptive networks (No. RSRE-MEMO-4148). Complex Syst 2:321–355

Brown M, Harris CJ (1994) Neurofuzzy adaptive modelling and control. Prentice Hall

Brownlie WR (1981) Prediction of flow depth and sediment discharge in open channels. California Institute of Technology, Pasadena, USA, Rep. KH-R-43A 232:1981

Buyukyildiz M, Kumcu SY (2017) An estimation of the suspended sediment load using adaptive network based fuzzy inference system, support vector machine and artificial neural network models. Water Resour Manag 31(4):1343–1359

Cannas B, Fanni A, Sias G, Tronci S, Zedda MK (2005) River flow forecasting using neural networks and wavelet analysis. Geophys Res Abstr 7:08651

Cannas B, Fanni A, See L, Sias G (2006) Data pre-processing for river flow forecasting using neural networks: wavelet transforms and data partitioning. Phys Chem Earth Parts A/B/C 31(18):1164–1171

Cevik A (2007) Genetic programming based formulation of rotation capacity of wide flange beams. J Constr Steel Res 63(7):884–893

Chien N, Wan Z (1999) Mechanics of sediment transport. American society of civil engineers. Reston, VA

Ch S, Anand N, Panigrahi BK, Mathur S (2013) Streamflow forecasting by SVM with quantum behaved particle swarm optimization. Neurocomputing 101:18–23

Chapelle O (2007) Training a support vector machine in the primal. Neural Comput 19(5):1155–1178

Chen XY, Chau KW (2016) A hybrid double feedforward neural network for suspended sediment load estimation. Water Resour Manag 30(7):2179–2194

Chen FL, Li FC (2010) Combination of feature selection approaches with SVM in creditscoring. Expert Syst Appl 37(7):4902–4909

Cheng CH, Chan CP, Yang JH (2018) A seasonal time-series model based on gene expression programming for predicting financial distress. Comput Intell Neurosci. https://doi.org/10.1155/2018/1067350

Chinnasamy P, Sood A (2020) Estimation of sediment load for Himalayan Rivers: Case study of Kaligandaki in Nepal. J Earth Syst Sci 129(1):1–18

Choubin B, Malekian A, Samadi S, Khalighi-Sigaroodi S, Sajedi-Hosseini F (2017) An ensemble forecast of semi-arid rainfall using large-scale climate predictors. Meteorol Appl 24(3):376–386

Choubin B, Darabi H, Rahmati O, Sajedi-Hosseini F, Kløve B (2018) River suspended sediment modelling using the CART model: a comparative study of machine learning techniques. Sci Total Environ 615:272–281

Cigizoglu HK, Alp M (2004) Rainfall-runoff modelling using three neural network methods. International conference on artificial intelligence and soft computing. Springer, Berlin, pp 166–171

Cigizoglu HK, Kisi Ö (2006) Methods to improve the neural network performance in suspended sediment estimation. J Hydrol 317(3–4):221–238

Cimen M (2008) Estimation of daily suspended sediments using support vector machines. Hydrol Sci J 53(3):656–666

Colby BR and Hembree CH (1955) Computation of total sedimentation discharge, Niobrara River, Nebraska. Water Supply Paper. Paper No. 1357, U.S. Geological Survey, Washington, D.C

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Systems 2(4):303–314

Daneshvar MRM, Bagherzadeh A (2012) Evaluation of sediment yield in PSIAC and MPSIAC models by using GIS at Toroq Watershed, Northeast of Iran. Front Earth Sci 6(1):83–94

Dawson CW, Wilby R (1998) An artificial neural network approach to rainfall-runoff modelling. Hydrol Sci J 43(1):47–66

Debao C (1993) Degree of approximation by superpositions of a sigmoidal function. Approx Theory Appl 9(3):17–28

Dibike YB, Solomatine DP (2001) River flow forecasting using artificial neural networks. Phys Chem Earth B 26(1):1–7

Dibike YB, Solomatine D, Abbott MB (1999) On the encapsulation of numerical-hydraulic models in artificial neural network. J Hydraul Res 37(2):147–161

Dragomiretskiy K, Zosso D (2013) Variational mode decomposition. IEEE Trans Signal Process 62(3):531–544

Drucker H, Burges CJ, Kaufman L, Smola AJ, Vapnik V (1997) Support vector regression machines. Advances in neural information processing systems. Springer, Berlin, pp 155–161

Duan W, Takara K (2020) Estimation of nutrient and suspended sediment loads in the Ishikari River. Impacts of climate and human activities on water resources and quality. Springer, Singapore, pp 127–159

Dutta S, Sen D (2018) Application of SWAT model for predicting soil erosion and sediment yield. Sustaine Water Resour Manag 4(3):447–468

Ebrahimi H, Jabbari E, Ghasemi M (2015) Application of the honey-bees mating programming (HBMP) algorithm to sediment concentration modelling. Hydrol Sci J 60(10):1853–1864

Ehteram M, Ghotbi S, Kisi O, Ahmed AN, Ahmed GH (2019) River suspended sediment prediction using improved anfis and ann models: comparative evaluation of the soft computing models. Water (In press)

Ehteram M, Ahmed AN, Latif SD, Huang YF, Alizamir M, Kisi O, El-Shafie A (2020) Design of a hybrid ANN multi-objective whale algorithm for suspended sediment load prediction. Environ Sci Pollut Res 28(2):1596–1611

Einstein HA (1950) The bed-load function for sediment transportation in open channel flows (No. 1488-2016-124615). US Department of Agriculture, Washington

Emamgholizadeh S, Demneh RK (2019) A comparison of artificial intelligence models for the estimation of daily suspended sediment load: a case study on the Telar and Kasilian rivers in Iran. Water Supply 19(1):165–178

Engelund F, Hansen E (1967) A monograph on sediment transport in alluvial streams. Technical University of Denmark, Copenhagen

Fadaee M, Mahdavi-Meymand A, Zounemat-Kermani M (2020) Suspended sediment prediction using integrative soft computing models: on the analogy between the butterfly optimization and genetic algorithms. Geocarto Int. https://doi.org/10.1080/10106049.2020.1753821

Fahlman S, Lebiere C (1989) The cascade-correlation learning architecture. Adv Neural Inf Process Syst 2:524–532

Fausett LV (1994) Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall, Englewood Cliffs

Fernandes FC, van Spaendonck RL, Burrus CS (2003) A new framework for complex wavelet transforms. IEEE Trans Signal Process 51(7):1825–1837

Ferreira C (2002) Gene expression programming in problem solving. Soft computing and industry. Springer, London, pp 635–653

Ferreira C, Gepsoft U (2008) What is gene expression programming. A new adaptive algorithm for solving problems 2002:87–129

Firat M, Güngör M (2008) Hydrological time-series modelling using an adaptive neuro-fuzzy inference system. Hydrol Process 22(13):2122–2132

Flood I, Kartam N (1994) Neural networks in civil engineering. I: Principles and understanding. J Comput Civil Eng 8(2):131–148

Frank E, Wang Y, Inglis S, Holmes G, Witten IH (1998) Using model trees for classification. Mach Learn 32(1):63–76

Freiwan M, Cigizoglu HK (2005) Prediction of total monthly rainfall in Jordan using feed forward backpropagation method. Fresenius Environ Bull 14(2):142–151

Fuller R, Fullér R (2000) Introduction to neuro-fuzzy systems. Springer Science and Business Media, Berlin

Funahashi KI (1989) On the approximate realization of continuous mappings by neural networks. Neural Netw 2(3):183–192

Fung G, Mangasarian OL (2003) Finite Newton method for Lagrangian support vector machine classification. Neurocomputing 55(1–2):39–55

Gallant SI, Gallant SI (1993) Neural network learning and expert systems. MIT Press, London

Gharde KD, Mahesh K, Mittal HK, Singh PK, Dahiphale PA (2015) Sediment yield modelling of Kal river in Maharashtra using artificial neural network model. Res J Recent Sci 4:120–130

Ghose DK (2018) Prediction of suspended sediment load using radial basis neural network. In: Bhateja V, CoelloCoello C, Satapathy S, Pattnaik P (eds) Intelligent engineering informatics. Advances in intelligent systems and computing. Springer, Singapore

Goodfellow IJ, Vinyals O, Saxe AM (2015) Qualitatively characterizing neural network optimization problems. In: 3rd International Conference on Learning Representations, ICLR 2015-Conference Track Proceedings, pp 1–20

Govindaraju RS (2000) Artificial neural networks in hydrology. II: hydrologic applications. J Hydrol Eng 5(2):124–137

Goyal MK (2014) Modeling of sediment yield prediction using M5 model tree algorithm and wavelet regression. Water Resour Manag 28(7):1991–2003

Goyal MK, Ojha CSP (2011) Estimation of scour downstream of a ski-jump bucket using support vector and M5 model tree. Water Resour Manag 25(9):2177–2195

Goyal MK, Ojha CSP, Singh RD, Swamee PK, Nema RK (2013) Application of ANN, fuzzy logic and decision tree algorithms for the development of reservoir operating rules. Water Resour Manag 27(3):911–925

Grubert JP (1995) Application of neural networks in stratified flow stability analysis. J Hydraul Eng 121(7):523–532

Gupta AK, Rudra RP, Gharabaghi B, Daggupati P, Goel PK, Shukla R (2018) Predicting the impact of drainage ditches upon hydrology and sediment loads using KINEROS 2 model: a case study in Ontario. Canad Biosyst Eng 60:1–1

Gupta D, Hazarika BB, Berlin M (2020) Robust regularized extreme learning machine with asymmetric Huber loss function. Neural Comput Appl 32:12971–12998

Haber E, Ruthotto L (2017) Stable architectures for deep neural networks. Inverse Prob 34(1):014004

Hamaamin YA, Nejadhashemi AP, Zhang Z, Giri S, Adhikari U, Herman MR (2019) Evaluation of neuro-fuzzy and Bayesian techniques in estimating suspended sediment loads. Sustain Water Resour Manag 5(2):639–654

Haque MZ, Rahim S, Abdullah MP, Embi AF, Elfithri R, Lihan T, Mokhtar M (2016) Predicting Sediment Load and Runoff in Geo WEPP Environment from Langat Sub Basin, Malaysia. Nat Environ Pollut Technol 15(3):1077

Hassanpour F, Sharifazari S, Ahmadaali K, Mohammadi S, Sheikhalipour Z (2019) Development of the FCM-SVR Hybrid Model for Estimating the Suspended Sediment Load. KSCE J Civil Eng 23(6):2514–2523

Haughton D, Oulabi S (1997) Direct marketing modeling with CART and CHAID. J Direct Market 11(4):42–52

Hazarika BB, Gupta D (2020) Density-weighted support vector machines for binary class imbalance learning. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05240-8

Hazarika BB, Gupta D, Berlin M (2020a) A coiflet LDMR and coiflet OB-ELM for river suspended sediment load prediction. Int J Environ Sci Technol. https://doi.org/10.1007/s13762-020-02967-8

Hazarika BB, Gupta D, Berlin M (2020b) Modeling suspended sediment load in a river using extreme learning machine and twin support vector regression with wavelet conjunction. Environ Earth Sci 79:1–15

Hazarika BB, Gupta D, Berlin M (2020c) A comparative analysis of artificial neural network and support vector regression for river suspended sediment load prediction. First international conference on sustainable technologies for computational intelligence. Springer, Singapore, pp 339–349

Heng S, Suetsugi T (2015) Regionalization of sediment rating curve for sediment yield prediction in ungauged catchments. Hydrol Res 46(1):26–38

Hill DA, Delaney LM, Roncal S (1997) A chi-square automatic interaction detection (CHAID) analysis of factors determining trauma outcomes. J Trauma Acute Care Surg 42(1):62–66

Himanshu SK, Pandey A, Yadav B (2017a) Assessing the applicability of TMPA-3B42V7 precipitation dataset in wavelet-support vector machine approach for suspended sediment load prediction. J Hydrol 550:103–117

Himanshu SK, Pandey A, Yadav B (2017b) Ensemble wavelet-support vector machine approach for prediction of suspended sediment load using hydro-meteorological data. J Hydrol Eng 22(7):05017006

Hong WC (2008) Rainfall forecasting by technological machine learning models. Appl Math Comput 200(1):41–57

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366

Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501

Ishikawa M (1996) Structural learning with forgetting. Neural Netw 9(3):509–521

Jaiyeola AT, Adeyemo J (2019) Performance comparison between genetic programming and sediment rating curve for suspended sediment prediction. Afr J Sci Technol Innov Dev 11(7):843–859

Jalalkamali A, Sedghi H, Manshouri M (2011) Monthly groundwater level prediction using ANN and neuro-fuzzy models: a case study on Kerman plain Iran. J Hydroinform 13(4):867–876

Jang JS, Sun CT (1995) Neuro-fuzzy modeling and control. Proc IEEE 83(3):378–406

Ji X, Lu J (2018) Forecasting riverine total nitrogen loads using wavelet analysis and support vector regression combination model in an agricultural watershed. Environ Sci Pollut Res 25(26):26405–26422

Jie LC, Yu ST (2011) Suspended sediment load estimate using support vector machines in Kaoping River basin. In: International Conference on Consumer Electronics, Communications and Networks (IEEE), XianNing, China, pp 16–18

Kalteh AM, Hjorth P, Berndtsson R (2008) Review of the self-organizing map (SOM) approach in water resources: analysis, modelling and application. Environ Model Softw 23(7):835–845

Karim MF, Kennedy JF (1990) Menu of coupled velocity and sediment-discharge relations for rivers. J Hydraul Eng 116(8):978–996

Karunanithi N, Grenney WJ, Whitly D, Bovee K (1994) Neural networks for river flow prediction. J Comput Civil Eng 8(2):201–220

Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. Appl Stat 29:119–127

Kaveh K, Kaveh H, Bui MD, Rutschmann P (2020) Long short-term memory for predicting daily suspended sediment concentration. Eng Comput. https://doi.org/10.1007/s00366-019-00921-y

Kayadelen C (2011) Soil liquefaction modelling by genetic expression programming and neuro-fuzzy. Expert Syst Appl 38(4):4080–4087

Khan MS, Coulibaly P (2006) Application of support vector machine in lake water level prediction. J Hydrol Eng 11(3):199–205

Khan MYA, Hasan F, Tian F (2019a) Estimation of suspended sediment load using three neural network algorithms in Ramganga River catchment of Ganga Basin India. Sustain Water Resour Manag 5(3):1115–1131

Khan MYA, Tian F, Hasan F, Chakrapani GJ (2019b) Artificial neural network simulation for prediction of suspended sediment concentration in the River Ramganga, Ganges Basin India. Int J Sediment Res 34(2):95–107

Kim TW, Valdés JB (2003) Nonlinear model for drought forecasting based on a conjunction of wavelet transforms and neural networks. J Hydrol Eng 8(6):319–328

Kişi Ö (2008) River flow forecasting and estimation using different artificial neural network techniques. Hydrol Res 39(1):27–40

Kisi O (2012) Modeling discharge-suspended sediment relationship using least square support vector machine. J Hydrol 456:110–120

Kisi O (2016) A new approach for modeling suspended sediment: evolutionary fuzzy approach. Hydrol Earth Syst Sci 58(3):587–599

Kisi O, Karahan ME, Şen Z (2006) River suspended sediment modelling using a fuzzy logic approach. Hydrol Process: Int J 20(20):4351–4362

Kisi O, Yaseen ZM (2019) The potential of hybrid evolutionary fuzzy intelligence model for suspended sediment concentration prediction. CATENA 174:11–23

Kisi O, Zounemat-Kermani M (2016) Suspended sediment modeling using neuro-fuzzy embedded fuzzy c-means clustering technique. Water Resour Manag 30(11):3979–3994

Korkobi T, Djemel M, Chtourou M (2008) Stability analysis of neural networks-based system identification. Model Simul Eng. https://doi.org/10.1155/2008/343940

Kumar D, Pandey A, Sharma N, Flügel WA (2015) Modeling suspended sediment using artificial neural networks and TRMM-3B42 version 7 rainfall dataset. J Hydrol Eng 20(6):C4014007

Lafdani EK, Nia AM, Ahmadi A (2013) Daily suspended sediment load prediction using artificial neural networks and support vector machines. J Hydrol 478:50–62

Leonard JA, Kramer MA, Ungar LH (1992) A neural network architecture that computes its own reliability. Comput Chem Eng 16(9):819–835

Li B, Cheng C (2014) Monthly discharge forecasting using wavelet neural networks with extreme learning machine. Sci China Technol Sci 57(12):2441–2452

Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22

Lin JY, Cheng CT, Chau KW (2006) Using support vector machines for long-term discharge prediction. Hydrol Sci J 51(4):599–612

Liu Y, Yang W, Yu Z, Lung I, Gharabaghi B (2015) Estimating sediment yield from upland and channel erosion at a watershed scale using SWAT. Water Resour Manag 29(5):1399–1412

Livni R, Shalev-Shwartz S, Shamir O (2014) On the computational efficiency of training neural networks. arXiv preprint arXiv:1410.1141

Lodge A, Yu XH (2014) Short term wind speed prediction using artificial neural networks. In: 2014 4th IEEE International Conference on Information Science and Technology, IEEE, pp 539–542

Lohani AK, Goel NK, Bhatia KS (2007) Deriving stage–discharge–sediment concentration relationships using fuzzy logic. Hydrol Sci J 52(4):793–807

Malik A, Kumar A, Piri J (2017) Daily suspended sediment concentration simulation using hydrological data of Pranhita River Basin, India. Comput Electron Agric 138:20–28

Malik A, Kumar A, Kisi O, Shiri J (2019) Evaluating the performance of four different heuristic approaches with Gamma test for daily suspended sediment concentration modeling. Environ Sci Pollut Res 26(22):22670–22687

Mangal R, Nori AV, Orso A (2019) Robustness of neural networks: a probabilistic and practical approach. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), IEEE, pp 93–96

Mareček J (2016) Usage of generalized regression neural networks in determination of the enterprise's future sales plan. Littera Scr 3:32–41

Martínez-Salvador A, Conesa-García C (2020) Suitability of the SWAT model for simulating water discharge and sediment load in a karst watershed of the semiarid Mediterranean Basin. Water Resour Manag. https://doi.org/10.1007/s11269-019-02477-4

Maxwell AE (1975) Limitations on the use of the multiple linear regression model. Br J Math Stat Psychol 28(1):51–62

McCarney-Castle K, Childress TM, Heaton CR (2017) Sediment source identification and load prediction in a mixed-use Piedmont watershed, South Carolina. J Environ Manag 185:60–69

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133

Mehr AD, Şorman AU (2018) Streamflow and sediment load prediction using linear genetic programming. Uludağ Univ J Fac Eng 23(2):323–332

Meshram SG, Ghorbani MA, Deo RC, Kashani MH, Meshram C, Karimi V (2019) New approach for sediment yield forecasting with a two-phase feedforward neuron network-particle swarm optimization model integrated with the gravitational search algorithm. Water Resour Manag 33(7):2335–2356

Mijwel MM (2018) Artificial neural networks advantages and disadvantages. Retrieved from Linked. https://www.linkedin.com/pulse/artificial-neuralnetworks-advantages-disadvantages-maad-m-mijwel. Accessed June 2020

Misra D, Oommen T, Agarwal A, Mishra SK, Thompson AM (2009) Application andanalysis of support vector machine based simulation for runoff and sedimentyield. Biosyst Eng 103(4):527–535

Moeeni H, Bonakdari H (2018) Impact of normalization and input on ARMAX-ANN model performance in suspended sediment load prediction. Water Resour Manag 32(3):845–863

Mohamed I, Shah I (2018) Suspended sediment concentration modeling using conventional and machine learning approaches in the Thames River, London Ontario. J Water Manag Model. https://doi.org/10.14796/JWMM.C453

Moisen GG (2008) Classification and regression trees. In: Jorgensen SE, Fath BD (eds) Encyclopedia of Ecology. Elsevier, Oxford, pp 582–588

Mondal SK (2011) A comparative study for prediction of direct runoff for a river basin using geomorphological approach and artificial neural networks. Appl Water Sci 2:1–13

Morgan RPC, Quinton JN, Smith RE, Govers G, Poesen JWA, Auerswald K, Styczen ME (1998) The European Soil Erosion Model (EUROSEM): a dynamic approach for predicting sediment transport from fields and small catchments. Earth Surf Process Landf 23(6):527–544

Mustafa MR (2016) Modelling daily suspended sediments of a hyper-concentrated river in Malaysia. Arpn J Eng Appl Sci 11(4):2141–2145

Nagy HM, Watanabe KAND, Hirano M (2002) Prediction of sediment load concentration in rivers using artificial neural network model. J Hydraul Eng 128(6):588–595

Naidu BSK (1999) Developing silt consciousness in the minds of hydropower engineers. In: Proceedings of 1st International Conference on Silting Problems in Hydropower Plants, India, pp 1–36

Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I. A discussion of principles. J Hydrol 10(3):282–290

Nayak PC, Sudheer KP, Rangan DM, Ramasastri KS (2004) A neuro-fuzzy computing technique for modeling hydrological time series. J Hydrol 291(1–2):52–66

Nhu VH, Khosravi K, Cooper JR, Karimi M, Kisi O, Pham BT, Lyu Z (2020) Monthly suspended sediment load prediction using artificial intelligence: testing of a new random subspace method. Hydrol Sci J 65(12):2116–2127

Nikoo M, Razavi SA, Hadzima-Nyarko M (2018) Artificial Neural Network Combined with Imperialist Competitive Algorithm for Determination of River Sediments. Fresenius Environ Bull 27(7):4658–4667

Nisbet R, Elder J, Miner G (2009) Handbook of statistical analysis and data mining applications. Academic Press

Nivesh S, Kumar P, Saran B, Sawant PN, Verma R (2019) Assessment of soft computing and statistical approaches for suspended sediment load estimation: Vamsadhara river basin, India. Pharm Innov J 8(2):693–702

Noori N, Kalin L (2016) Coupling SWAT and ANN models for enhanced daily streamflow prediction. J Hydrol 533:141–151

Noori R, Karbassi AR, Moghaddamnia A, Han D, Zokaei-Ashtiani MH, Farokhnia A, Gousheh MG (2011) Assessment of input variables determination on the SVM modelperformance using PCA, Gamma test, and forward selection techniques for monthly streamflow prediction. J Hydrol 401(3–4):177–189

Noori R, Deng Z, Kiaghadi A, Kachoosangi FT (2015) How reliable are ANN, ANFIS, and SVM techniques for predicting longitudinal dispersion coefficient in natural rivers? J Hydraul Eng 142(1):04015039

Nourani V, Andalib G (2015a) Daily and monthly suspended sediment load predictions using wavelet based artificial intelligence approaches. J Mt Sci 12(1):85–100

Nourani V, Andalib G (2015b) Wavelet based artificial intelligence approaches for prediction of hydrological time series. Australasian conference on artificial life and computational intelligence. Springer, Cham, pp 422–435

Nourani V, Alami MT, Aminfar MH (2009) A combined neural-wavelet model for prediction of Ligvanchai watershed precipitation. Eng Appl Artif Intell 22(3):466–472

Nourani V, Davanlou Tajbakhsh A, Molajou A, Gokcekus H (2019) Hybrid wavelet-M5 model tree for rainfall-runoff modeling. J Hydrol Eng 24(5):04019012

Nourani V, Alizadeh F, Roushangar K (2016) Evaluation of a two-stage SVM and spatial statistics methods for modeling monthly river suspended sediment load. Water Resour Manag 30(1):393–407

Nourani V, Andalib G, Dąbrowska D (2017) Conjunction of wavelet transform and SOM-mutual information data pre-processing approach for AI-based Multi-Station nitrate modeling of watersheds. J Hydrol 548:170–183

Ocampo-Duque W, Schuhmacher M, Domingo JL (2007) A neural-fuzzy approach to classify the ecological status in surface waters. Environ Pollut 148(2):634–641

Oeurng C, Sauvage S, Sánchez-Pérez JM (2011) Assessment of hydrology, sediment and particulate organic carbon yield in a large agricultural catchment using the SWAT model. J Hydrol 401(3–4):145–153

Olyaie E, Banejad H, Chau KW, Melesse AM (2015) A comparison of various artificial intelligence approaches performance for estimating suspended sediment load of river systems: a case study in United States. Environ Monit Assess 187(4):189

Ouedraogo I, Defourny P, Vanclooster M (2019) Application of random forest regression and comparison of its performance to multiple linear regression inmodeling groundwater nitrate concentration at the African continent scale. Hydrol J 27(3):1081–1098

Özger M, Kabataş MB (2015) Sediment load prediction by combined fuzzy logic-wavelet method. J Hydroinf 17(6):930–942

Panagoulia D, Tsekouras GJ, Kousiouris G (2017) A multi-stage methodology for selecting input variables in ANN forecasting of river flows. Glob Nest J 19:49–57

Partal T, Cigizoglu HK (2008) Estimation and forecasting of daily suspended sediment data using wavelet–neural networks. J Hydrol 358(3–4):317–331

Partovian A, Nourani V, Alami MT (2016) Hybrid denoising-jittering data processing approach to enhance sediment load prediction of muddy rivers. J Mt Sci 13(12):2135–2146

Pektaş AO, Doğan E (2015) Prediction of bed load via suspended sediment load using soft computing methods. Geofizika 32(1):27–46

Peng X (2010a) Primal twin support vector regression and its sparse approximation. Neurocomputing 73(16–18):2846–2858

Peng X (2010b) TSVR: an efficient twin support vector machine for regression. Neural Netw 23(3):365–372

Quinlan JR (1992) Learning with continuous classes. Aust Jt Conf Artif Intell 92:343–348

Raghuwanshi NS, Singh R, Reddy LS (2006) Runoff and sediment yield modeling using artificial neural networks: Upper Siwane River, India. J Hydrol Eng 11(1):71–79

Rahgoshay M, Feiznia S, Arian M, Hashemi SAA (2018) Modeling daily suspended sediment load using improved support vector machine model and genetic algorithm. Environ Sci Pollut Res 25(35):35693–35706

Rajaee T (2010) Wavelet and neuro-fuzzy conjunction approach for suspended sediment prediction. Clean: Soil, Air, Water 38(3):275–286

Rajaee T (2011) Wavelet and ANN combination model for prediction of daily suspended sediment load in rivers. Sci Total Environ 409(15):2917–2928

Rajaee T, Mirbagheri SA, Zounemat-Kermani M, Nourani V (2009) Daily suspended sediment concentration simulation using ANN and neuro-fuzzy models. Sci Total Environ 407(17):4916–4927

Rajaee T, Nourani V, Zounemat-Kermani M, Kisi O (2010) River suspended sediment load prediction: application of ANN and wavelet conjunction model. J Hydrol Eng 16(8):613–627

Ramezani F, Nikoo M, Nikoo M (2015) Artificial neural network weights optimization based on social-based algorithm to realize sediment over the river. Soft Comput 19(2):375–387

Rashidi S, Vafakhah M, Lafdani EK, Javadi MR (2016) Evaluating the support vector machine for suspended sediment load forecasting based on gamma test. Arab J Geosci 9(11):583

Rastogi R, Anand P, Chandra S (2020) Large-margin distribution machine-based regression. Neural Comput Appl 32(8):3633–3648

Rezaie-balf M, Naganna SR, Ghaemi A, Deka PC (2017) Wavelet coupled MARS and M5 model tree approaches for groundwater level forecasting. J Hydrol 553:356–373

Rezapour OM, Shui LT, Dehghani AA (2010) Review of genetic algorithm model for suspended sediment estimation. Aust J Basic Appl Sci 4(8):3354–3359

Rijn LCV (1984) Sediment transport, Rijn 1984, part II: suspended load transport. J Hydraul Eng 110(11):1613–1641

Rojas R (1996) Neural networks: a systematic introduction. Chap 1:3–28

Roushangar K, Hosseinzadeh S, Shiri J (2016) Local vs. cross station simulation of suspended sediment load in successive hydrometric stations: heuristic modeling approach. J Mt Sci 13(10):1773–1788

Rumelhart DE, Hinton GE, Williams RJ (1988) Learning representations by back-propagating errors. Cognit Model 5(3):1

Sadeghpour Haji M, Mirbagheri SA, Javid AH, Khezri M, Najafpour GD (2014) Suspended sediment modelling by SVM and wavelet. Građevinar 66(03):211–223

Sahoo GB, Ray C (2006) Flow forecasting for a Hawaii stream using rating curves and neural networks. J Hydrol 317(1–2):63–80

Salih SQ, Sharafati A, Khosravi K, Faris H, Kisi O, Tao H, Yaseen ZM (2019) River suspended sediment load prediction based on river discharge information: application of newly developed data mining models. Hydrol Sci J 65:624–637

Samantaray S, Sahoo A, Ghose DK (2020) Prediction of sedimentation in an arid watershed using BPNN and ANFIS. In: Fong S, Dey N, Joshi A (eds) ICT analysis and applications lecture notes in networks and systems. Springer, Singapore

Samet K, Hoseini K, Karami H, Mohammadi M (2019) Comparison between soft computing methods for prediction of sediment load in rivers: Maku dam case study. Iran J Sci Technol Trans Civil Eng 43(1):93–103

Sarangi A, Bhattacharya AK (2005) Comparison of artificial neural network and regression models for sediment loss prediction from Banha watershed in India. Agric Water Manag 78(3):195–208

Sarangi A, Madramootoo CA, Enright P, Prasher SO, Patel RM (2005) Performance evaluation of ANN and geomorphology-based models for runoff and sediment yield prediction for a Canadian watershed. Current Sci 89:2022–2033

Sari V, dos Reis Castro NM, Pedrollo OC (2017) Estimate of suspended sediment concentration from monitored data of turbidity and water level using artificial neural networks. Water Resour Manag 31(15):4909–4923

Sattari MT, Rezazadeh JA, Safdari F, Ghahramanian F (2016) Performance evaluation of M5 tree model and support vector regression methods in suspended sediment load modeling. J Soil Water Resour Conserv 6(1):109–124

Shamaei E, Kaedi M (2016) Suspended sediment concentration estimation by stacking the genetic programming and neuro-fuzzy predictions. Appl Soft Comput 45:187–196

Sharghi E, Nourani V, Najafi H, Gokcekus H (2019) Conjunction of a newly proposed emotional ANN (EANN) and wavelet transform for suspended sediment load modeling. Water Supply 19(6):1726–1734

Sharma D, Lie TT (2012) Wind speed forecasting using hybrid ANN-Kalman filter techniques. In 2012 10th International Power and Energy Conference (IPEC), IEEE, pp 644–648

Shiau JT, Chen TJ (2015) Quantile regression-based probabilistic estimation scheme for daily and annual suspended sediment loads. Water Resour Manag 29(8):2805–2818

Shiri J, Kişi Ö (2011) Estimation of daily suspended sediment load by using wavelet conjunction models. J Hydrol Eng 17(9):986–1000

Shiri N, Shiri J, Nourani V, Karimi S (2020) Coupling wavelet transform with multivariate adaptive regression spline for simulating suspended sediment load: independent testing approach. ISH J Hydraul Eng. https://doi.org/10.1080/09715010.2020.1801528

Shrestha DL, Solomatine DP (2006) Experiments with AdaBoost. RT, an improved boosting scheme for regression. Neural Comput 18(7):1678–1710

Singh KK, Pal M, Singh VP (2010) Estimation of mean annual flood in Indian catchments using backpropagation neural network and M5 model tree. Water Resour Manag 24(10):2007–2019

Sivapragasam C, Muttil N (2005) Discharge rating curve extension: a new approach. Water Resour Manag 19(5):505–520

Sivapragasam C, Liong SY, Pasha MFK (2001) Rainfall and runoff forecasting with SSA–SVM approach. J Hydroinf 3(3):141–152

Smith J, Eli RN (1995) Neural-network models of rainfall-runoff process. J Water Resour Plan Manag 121(6):499–508

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Specht DF (1990) Probabilistic neural networks. Neural Netw 3(1):109–118

Splines MAR (2013) Retrieved from StatSoft: http://www.statsoft.com/Textbook. Accessed June 2020

Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300

Tabatabaei M, Jam AS, Hosseini SA (2019) Suspended sediment load prediction using non-dominated sorting genetic algorithm II. Int Soil Water Conserv Res 7(2):119–129

Tachi SE, Bouguerra H, Derdous O, Djabri L, Benmamar S (2020) Estimating suspended sediment concentration at different time scales in Northeastern Algeria. Appl Water Sci 10:1–8

Talebi A, Mahjoobi J, Dastorani MT, Moosavi V (2017) Estimation of suspended sedimentload using regression trees and model trees approaches (Case study: Hyderabad drainage basin in Iran). ISH J Hydraul Eng 23(2):212–219

Tantanee S, Patamatammakul S, Oki T, Sriboonlue V, Prempree T (2005) Coupled wavelet-autoregressive model for annual rainfall prediction. J Environ Hydrol 13:124–146

Tao H, Keshtegar B, Yaseen ZM (2019) The feasibility of integrative radial basis M5Tree predictive model for river suspended sediment load simulation. Water Resour Manag 33(13):4471–4490

Tarar Z, Ahmad S, Ahmad I, Majid Z (2018) Detection of sediment trends using wavelet transforms in the Upper Indus river. Water 10(7):918

Taşar B, Kaya YZ, Varçin H, Üneş F, Demirci M (2017) Forecasting of suspended sediment in rivers using artificial neural networks approach. Int J Adv Eng Res Sci 4(12):79–84

Tayfur G, Guldal V (2006) Artificial neural networks for estimating daily total suspended sediment in natural streams. Hydrol Res 37(1):69–79

Tayfur G, Ozdemir S, Singh VP (2003) Fuzzy logic algorithm for runoff-induced sediment transport from bare soil surfaces. Adv Water Resour 26(12):1249–1256

Teodorescu L, Sherwood D (2008) High energy physics event selection with gene expression programming. Comput Phys Commun 178(6):409–419

Tfwala SS, Wang YM (2016) Estimating sediment discharge using sediment rating curves and artificial neural networks in the Shiwen River, Taiwan. Water 8(2):53

Thelwall M, Kousha K (2017) researchgate versus google scholar: which finds more early citations? Scientometrics 112(2):1125–1131

Timofeev R (2004) Classification and regression trees (CART) theory and applications. Humboldt University, Berlin, Wadsworth Stat. Ser. x, p 358

Tokar AS, Johnson PA (1999) Rainfall-runoff modeling using artificial neural networks. J Hydrol Eng 4(3):232–239

Toriman E, Jaafar O, Maru R, Arfan A, Ahmar AS (2018) Daily suspended sediment discharge prediction using multiple linear regression and artificial neural network. J Phys 954(1):012030

Tripathi S, Srinivas VV, Nanjundiah RS (2006) Downscaling of precipitation for climate change scenarios: a support vector machine approach. J hydrol 330(3–4):621–640

Tsykin E (1984) Multiple nonlinear regressions derived with choice of free parameters. Appl Math Model 8(4):288–292

Ulke A, Tayfur G, Ozkul S (2017) Investigating a suitable empirical model and performing regional analysis for the suspended sediment load prediction in major rivers of the Aegean Region, Turkey. Water Resour Manag 31(3):739–764

Üneş F, Karaeminoğullari AB, Taşar B (2020) Forecasting of river sediment amount using machine model. Int J Environ Agric Biotechnol 5(1):9–15

Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Networks 10(5):988–999

Vapnik V (2000) The nature of statistical learning theory. Red Bank. Springer, Berlin

Venkatesan D, Kannan K, Saravanan R (2009) A genetic algorithm-based artificial neural network model for the optimization of machining processes. Neural Comput Appl 18(2):135–140

Wang X, Sideratos G, Hatziargyriou, N., & Tsoukalas, L. H. (2004, September). Wind speed forecasting for power system operational planning. In: 2004 International Conference on Probabilistic Methods Applied to Power Systems, IEEE, pp 470–474

Werbos PJ (1974) New tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Dissertation, Harvard University.

Whitley D (1994) A genetic algorithm tutorial. Stat Comput 4(2):65–85

Witten IH, Frank E (2016) Data mining-practical machine learning tools and techniques. Morgan Kaufmann, San Francisco

Wu Z, Huang NE (2009) Ensemble empirical mode decomposition: a noise-assisted data analysis method. Adv Adapt Data Anal 1(01):1–41

Yadav A, Satyannarayana P (2019) Multi-objective genetic algorithm optimization of artificial neural network for estimating suspended sediment yield in Mahanadi River basin India. Int J River Basin Manag. https://doi.org/10.1080/15715124.2019.1705317

Yadav A, Chatterjee S, Equeenuddin SM (2018) Suspended sediment yield estimation using genetic algorithm-based artificial intelligence models: case study of Mahanadi River India. Hydrol Scie J 63(8):1162–1182

Yadav A, Chatterjee S, Equeenuddin SM (2020) Suspended sediment yield modeling in Mahanadi River, India by multi-objective optimization hybridizing artifilacial intelligence algorithms. Int J Sediment Res 36:76–91

Yaseen ZM, El-Shafie A, Afan HA, Hameed M, Mohtar WHMW, Hussain A (2016) RBFNN versus FFNN for daily river flow forecasting at Johor River Malaysia. Neural Comput Appl 27(6):1533–1542

Yilmaz B, Aras E, Kankal M, Nacar S (2019) Prediction of suspended sediment loading by means of hybrid artificial intelligence approaches. Acta Geophys 67(6):1693–1705

Yin S, Tang D, Jin X, Chen W, Pu N (2016) A combined rotated general regression neural network method for river flow forecasting. Hydrol Sci J 61(4):669–682

Yitian LI, Gu RR (2003) Modeling flow and sediment transport in a river system using an artificial neural network. Environ Manag 31(1):0122–0134

Yunkai L, Yingjie T, Zhiyun O, Lingyan W, Tingwu X, Peiling Y, Huanxun Z (2010) Analysis of soil erosion characteristics in small watersheds with particle swarm optimization, support vector machine, and artificial neuronal networks. Environ Earth Sci 60(7):1559–1568

Zarris D, Lykoudi E, Panagoulia D (2006) Assessing the impacts of sediment yield on the sustainability of major hydraulic systems. Int Conf Protect Restor Environ 8:3–7

Zarris D, Vlastara M, Panagoulia D (2011) Sediment delivery assessment for a transboundary Mediterranean catchment: the example of Nestos River catchment. Water Resour Manag 25(14):3785

Zende AM, Nagarajan R (2015) Sediment yield estimate of river Basin using SWAT Model in semi-arid region of peninsular india. In: Lollino G, Arattano M, Rinaldi M, Giustolisi O, Marechal JC, Grant G (eds) Engineering geology for society and territory. Springer, Cham

Zhang B, Govindaraju RS (2003) Geomorphology-based artificial neural networks (GANNs) for estimation of direct runoff over watersheds. J Hydrol 273(1–4):18–34

Zhang Q, Stanley SJ (1997) Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modeling. Water Res 31(9):2340–2350

Zhang X, Yang Y (2020) Suspended sediment concentration forecast based on CEEMDAN-GRU model. Water Supply 20(5):1787–1798

Zhang J, Zhao Y, Xiao W (2015) Multi-resolution cointegration prediction for runoff and sediment load. Water Resour Manag 29(10):3601–3613

Zhang H, Weng TW, Chen PY, Hsieh CJ, Daniel L (2018) Efficient neural network robustness certification with general activation functions. Advances in neural information processing systems. Springer, Berlin, pp 4939–4948

Zhang M, Huang X, Li Y, Sun H, Zhang J, Huang B (2020) Improved continuous wavelet transform for modal parameter identification of long-span bridges. Shock Vib. https://doi.org/10.1155/2020/4360184

Zhao DF, Wang M, Zhang JS, Wang XF (2002) A support vector machine approach for short term load forecasting. Proc CSEE 22(4):26–30

Zhu ML, Fujita M, Hashimoto N (1994) Application of neural networks to runoff prediction. Stochastic and statistical methods in hydrology and environmental engineering. Springer, Dordrecht, pp 205–216

Zhu YM, Lu XX, Zhou Y (2007) Suspended sediment flux modeling with artificial neural network: an example of the Longchuanjiang River in the Upper Yangtze Catchment, China. Geomorphology 84(1–2):111–125

Zhu S, Zhou J, Ye L, Meng C (2016) Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze River, China. Environ Earth Sci 75(6):531

Zounemat-Kermani M, Teshnehlab M (2008) Using adaptive neuro-fuzzy inference system for hydrological time series prediction. Appl Soft Comput 8(2):928–936

Zounemat-Kermani M, Mahdavi-Meymand A, Alizamir M, Adarsh S, MundherYaseen Z (2020) On the complexities of sediment load modeling using integrative machine learning: an application to the great river of Loíza in Puerto Rico. J Hydrol 585:124759