Global NEST Journal, Vol 26, No 4, 05880

An efficient flood forecasting model using an optimal deep belief network

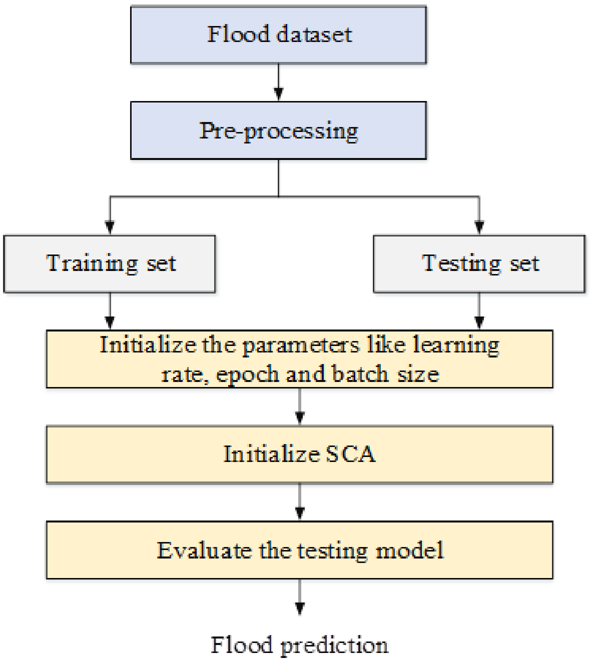
Bhaggiaraj S.[1], Jagadeesh M.[2]*, Krishnamoorthy M.[3] and Kumar R. D.[4]
[1]Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, 641022, India
[2]Department of Computing Technologies, SRM institute of Science and technology, Kattankulathur - 603203, Tamilnadu, India.
[3]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, 600123, India
[4]Department: Electronics and Communication Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602117, India

Received: 05/03/2024, Accepted: 18/04/2024, Available online: 27/04/2024

*to whom all correspondence should be addressed: e-mail: mjagadeesh.cse@gmail.com

https://doi.org/10.30955/gnj.005880

Graphical abstract


Flood dataset
Pre-processing
Training set
Testing set
Initialize the parameters like learning rate, epoch and batch size
Initialize SCA
Evaluate the testing model
Flood prediction

Abstract

Floods inflict significant damage globally each year, underscoring the importance of accurate and timely flood prediction to mitigate property loss and life. Precise flood prediction provides governments with crucial preemptive alerts regarding potential flood disasters, allowing timely evacuations and lifesaving measures. Although various ML (machine learning) models have shown improved performance compared to traditional statistical models in flood prediction, they often overlook the spatial features crucial to understanding the generation of floods. deep learning (DL) is used in flood prediction to enhance the promptness and efficiency of flood level predictions. This work presents an optimized DL model to forecast floods using time series data. Initially, the data set was cleaned and normalized by linear interpolation. Then, the DL model optimal deep belief network (ODBN) is utilized for flood forecasting time series prediction. ODBN is the integration of DBN and Sinh-Cosh algorithm (SCA). The experimental analysis is carried out on the real-time dataset and achieved better MSE and RMSE values of 0.75 and 0.94 respectively. The findings suggest that the use of an ODBN is an effective method of accurately forecasting floods.

Keywords: Flood prediction, time series data, deep learning, optimal deep belief network

1. Introduction

Floods are commonly influenced by a multitude of factors, including rainfall, evaporation, exposure to sunlight, surface conditions, and air movement in the atmosphere. Hakim *et al*. (2023) exhibit complex characteristics characterized by strong nonlinearity and high levels of uncertainty. Flooding constitutes a significant portion, approximately 84%, of global natural disaster deaths. Chenmin *et al* (2024) dicusssed various nations are grappling with recurrent floods that result in substantial direct economic losses of around US$ 60 billion annually, in addition to numerous casualties and injuries. In recent times, the increasing impacts of climate variation and changes in socioeconomic conditions have increased the frequency and intensity of floods. Consequently, this has imposed a pressing need on managers and engineers to delineate landscapes in both temporal and spatial dimensions, assessing the likelihood of flood events.

Farahmand *et al* (2023) implement the initial step in flood management schemes involves conducting an analysis of flood hazards for flood-prone areas. Linh *et al* (2021) analysis aims to determine landscapes with a high or very high probability of experiencing floods, providing crucial guidance for the development of effective management plans and the allocation of resources for the response to floods. Surendran R *et al* (2023) achieve this, it is imperative to employ robust and reliable tools that allow engineers to accurately assess the anticipated time, location, and future flood extent. Precise and timely flood prediction facilitates effective decision-making for flood control and mitigating potential losses. Currently, there are several methodologies to predict floods.

Onen *et al* (2017) find the conventional statistical approaches operate on the fundamental principles of flood formation. Jain *et al*. (2018) approaches demonstrate considerable predictive potential across a spectrum of flooding strategies, they frequently require extensive hydrological monitoring databases. The computational demands of these models hinder short-term predictions and require significant resources and time for development. Like numerical and statistical approaches, data-driven approaches have a long history in flood modelling. Han *et al* (2017) dicussed the deep learning (DL) has emerged as a prominent datadriven approach and it is used for predicting floods using factors like wind, rainfall, and temperature. Recent advances in DL models have improved the performance of different researchers to improve flood prediction precision.

Piadeh *et al* (2022), Nevo *et al* (2022), and Das *et al* (2022) introduced machine learning (ML), these DL models possess the ability to derive patterns from extensive data sets. Identification and forecasting of hydrological disasters employing DL models are closely linked to the availability of sufficiently large historical datasets. These substantial historical data are crucial to present well-informed strategic decisions for the future. Bentivoglio *et al* (2022), Zhong *et al* (2023), Luppichini *et al* (2022) and Ebtehaj *et al* (2022) discussed time series predictive approaches based on DL models used in hydrology have exhibited superior performance and cost-effectiveness. Some of the DL approaches such as convolutional neural network (CNN), recurrent neural network (RNN), and long-short-term memory (LSTM) are utilized in the construction of hydrological models. The foremost contributions of the work are:

- Present an automated, optimized deep learning-based model for flood forecasting in time-series data.
- To introduce an algorithm optimal deep belief network (ODBN) for flood forecasting by considering MSE as fitness.
- Execute different measures to analyze the performance of the suggested flood forecasting model.

The scope of optimized DL model is utilised in flood prediction to increase the promptness and efficiency of flood level predictions. The objective of the optimized DL model is to forecast floods using time series data while considering different parameters. The following sections are as follows: Section 2 provides an overview of existing flood prediction methods. Section 3 outlines the proposed flood prediction, while Section 4 presents the simulation results, and finally, Section 6 offers concluding remarks for the paper.

2. Literature survey

Hu *et al*. (2019) developed LSTM with ROM (reduced order model) to extract spatiotemporal features of the flood. Here, dimensionalityreducing approaches such as singular value decomposition (SVD) and proper orthogonal decomposition (POD) were performed. Prescriptive analysis was performed to estimate flood uncertainty on the Okushiri tsunami test data. Chen *et al*. (2022) employed ConvLSTM to capture spatio-temporal features inherent in hydrological information. Here, the designated area was partitioned into grids using data from the stations, longitude, latitude, and encompassing rainfall and discharge were amalgamated into tensors based on the station coordinates. The hydrological data was obtained in China and the peak discharge and arrival time values obtained were less than 20% and 30%.

Panahi *et al*. (2021) presented CNN with an RNN model for spatial explicit identification and probability of flash flood mapping. SWARA (stepwise weight assessment ratio analysis) was utilized to investigate spatial relations. A geospatial data set was considered and the precision and recall values achieved were 78.1% and 80.2%, respectively. Zou *et al*. (2023) developed a deep autoregressive recurrent model (DAR) model to forecast floods using time series data. Unlike the prevalent approach to generate deterministic flood predictions, it was essential to recognize that the hydrological properties of a basin constitute a complex and nonlinear system influenced by numerous terms. Therefore, addressing this complexity requires the adoption of probabilistic methodologies in modelling flood predictions. The accuracy of the peak flow prediction was approximately 90%.

Surendran R *et al*. (2023) presented deep neural network (DNN) to predict the occurrence of floods for rainfall and temperature. The DNN performance of the DNN was compared over different ML models and achieved better accuracy and recall values of 91.1% and 93%, respectively. The findings were based solely on monsoon parameters in the period prior to the occurrence of the flood. Löwe *et al*. (2021) presented the UFlood model to predict urban pluvial floods. This UFlood model was trained to extract topographic and hyetograph data. This existing approach demonstrated accurate water depth prediction for numerous rain factors and effectively discerned scenarios where flooding was not expected. However, the heightened prediction error observed highlighted the vulnerability in capturing the temporal dynamics of certain factors of the rain.

Lei *et al*. (2021) developed two DL models CNN and RNN for urban flood forecasting in Seoul in the country of South Korea. Here, the CNN model achieved better AUC and RMSE values of 84% and 0.16; then, the CNN model achieved better AUC and RMSE values of 82% and 0.18; it was consistently found that the terrain ruggedness index emerged as the most crucial predictor, with slope and elevation following in significance. Hosseiny and Hossein *et al*. (2023) Santhanaraj R. K *et al*. (2023) developed *UNet$_{river}$* a model for identifying river automatically identifying rever geometry and predicting the depth of the river. This existing work was provided with a combined image containing dual bands representing input as flood discharge and ground elevation, with the resulting output indicating the depth of the water. The model was analyzed in the Green River segment in Utah State. The *UNet_{river}*model achieved better training and validation losses of 0.0006 and 0.0012.

Surendran R. *et al.* (2021) Previous studies have incorporated various DL models to improve flood prediction performance. However, these models do not consider the optimal feature extraction and flood prediction performance. Additionally, the DL models employed in these studies exhibited overfitting problems. Therefore, the proposed work addresses these shortcomings by utilizing ODBN for optimal feature extraction and prediction for improved weather forecasting.

3. Proposed methodology

The surge in rainfall poses multiple challenges in various states, especially in urban areas, where sewer systems often struggle to cope with a substantial influx of water in a short time frame. Concurrently, conventional flood prediction outcomes prove unreliable for intricate occurrences and struggle with large volumes of data. To address these limitations and improve the effectiveness of traditional flood prediction models, this work presents an optimal DL model for controlling floods. Figure 1 defines the framework of the proposed flood prediction model.

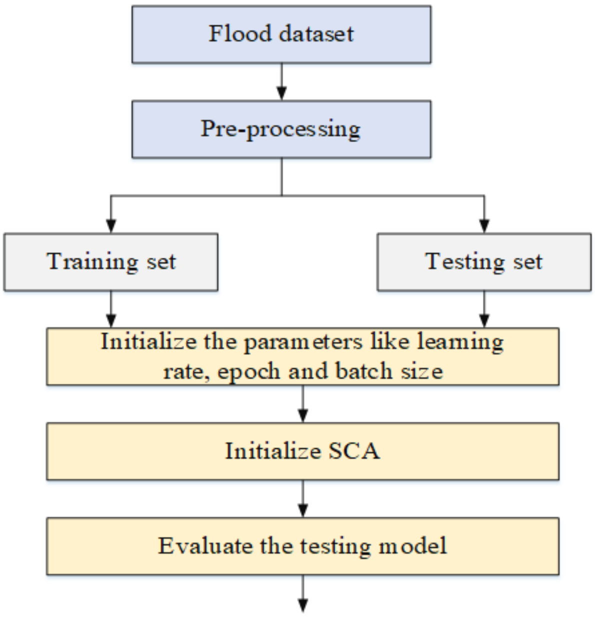

Flood dataset
Pre-processing
Training set
Testing set
Initialize the parameters like learning rate, epoch and batch size
Initialize SCA
Evaluate the testing model

Figure 1. Framework of the proposed flood prediction model

3.1. Preprocessing

Initially, in the pre-processing stages, the processes like missing data filling and normalization processes are carried out. The linear interpolation method is selected to fill in missing values due to the gradual difference observed in the time series. The expression to compute the linear interpolation is given as

$$y_l = \frac{t_{l+1} - t_l}{t_{l+1} - t_{l-1}} y_{l-1} + \frac{t_l - t_l - 1}{t_{l+1} - t_{l-1}} y_{l+1} \quad (1)$$

where y_l and t_l are the range and information of missing value; y_{l-1} and y_{l+1} are the final and next known ranges; t_{l-1} and t_{l+1} are the before and after the missing values.

To address the issue of varying data ranges among different flood prediction variables, it is crucial to normalize the raw data before feeding them into the DL model. A widely adopted normalization method is max-min normalization, which involves mapping all data in the range of 0 to 1. The normalization equation is expressed as

$$z' = \frac{z - z_{mi}}{z_{ma} - z_{mi}} \quad (2)$$

where Z and Z′ are the original and the normalized data; Z_{ma} and Z_{mi} are the maximum and minimum values.

3.2. Feature extraction

The ODBN is structured with a series of stacked restricted Boltzmann machine (RBM) layers, serving as fundamental components of the network, as shown in Figure 2. Each RBM comprises two layers of neurons: a visible layer (VL) and a hidden layer (HL). Bi-directional connections exist between every node in the VL and HL.

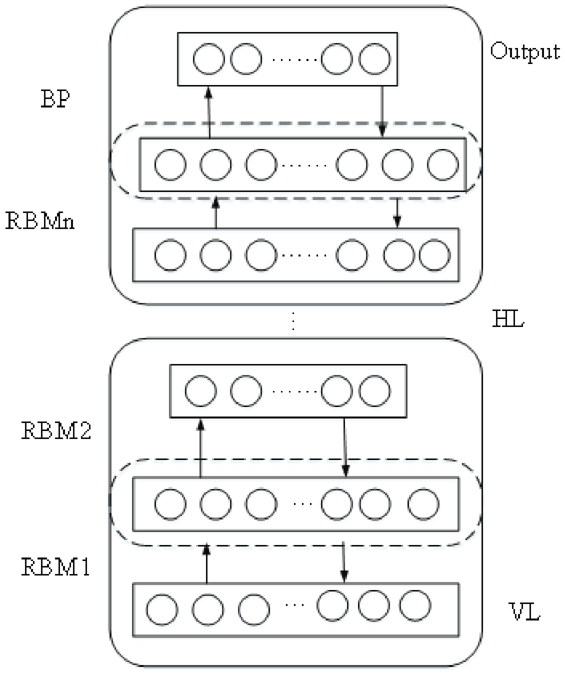


Figure 2. Structure of DBN

The training process involves unsupervised learning for every RBM, referred to as the pre-training phase. During RBM training, the primary objective is to determine initial parameters, including weights and biases, by maximizing likelihood estimation. This optimization aims to reconstruct the training samples effectively, forming the foundation for subsequent stages in network operation. The structure of RBM is shown in Figure 3. The HL and VL neurons have the ranges of $h \in \{0,1\}^M$ and $v \in \{0, 1\}^N$. The energy term of the integrated HL-VL is given as:

$$En\,(v,h) = -\lambda^T v - \alpha^T h - v^T uh \tag{3}$$

$$En\,(v,h) = -\sum_{j=1}^{N}\lambda_j v_j - \sum_{k=1}^{M}\alpha_k h_k - \sum_{j=1}^{N}\sum_{k=1}^{M} u_{jk} h_k v_j \tag{4}$$

where λj and *αk* are the biases of HL and VL neurons; *ujk* is the weight among VL (*vj*) and HL (*hj*) neurons. The probability joint distribution (*v*, *h*) is given as:

$$P\,(v,h) = \frac{\exp - En\,(v,h)}{\sum_{v,h}\exp - En\,(v,h)} \tag{5}$$

The term Σv, hexp−En (v, h) is the normalized term for every configuration of VL and HL neurons. The conditional probability distribution of HL and VL neurons of the *vj* and *hj* are given as:

$$p\,(h_k = 1\,|\,v) = \sigma\left(\alpha_k + \sum_{j=1}^{N} v_j u_{jk}\right) \tag{6}$$

$$p\,(v_j = 1\,|\,h) = \sigma\left(\lambda_k + \sum_{k=1}^{M} h_k u_{jk}\right) \tag{7}$$

where σ is the sigmoid function. The RBM is based on the parameters like weights u and the biases like *λ* and α. These parameters are estimated using the maximum log like-lihood (LL) and it is given as:

$$\log LL(\theta) = \sum_m \log p(V_m) \tag{8}$$

where θ = {λ, α, *u*} and *m* is the training instances. The gradient of the LL based on the parameters of the model is given as:

$$\frac{\partial \log p(v)}{\partial \lambda_j} = v_j - \sum_v p(v) v_j = En[v_j]_a - En[v_j]_r \tag{9}$$

$$\frac{\partial \log p(v)}{\partial \alpha_j} = p(h_k = 1\,|\,v) - \sum_v p(v) p(h_k = 1\,|\,v) = En[h_k]_a - En[h_k]_r \tag{10}$$

$$\frac{\partial \log p(v)}{\partial u_{jk}} = p(h_k = 1\,|\,v) v_j - \sum_v p(v) p(h_k = 1\,|\,v) v_j = En[v_j h_k]_a - En[v_j h_k]_r \tag{11}$$

where *En* [] a and *En* [] r are the actual and reconstructed data.

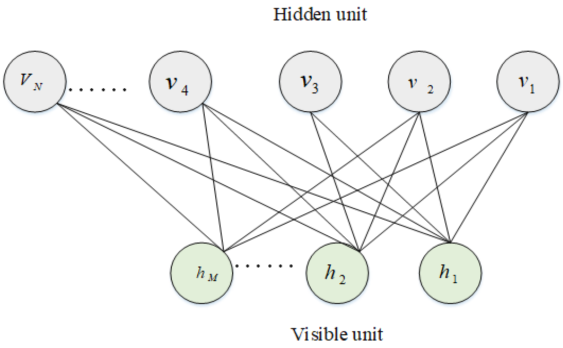


Figure 3. The Structure of RBM

The learning variables of the network are indicated by the ϕ (momentum) and β (learning rate) and they are given by the following expressions:

$$\Delta\lambda_j = \varphi\lambda_j + \beta En[v_j]_a - En[v_j]_r \tag{12}$$

$$\Delta\alpha_j = \varphi\alpha_j + \beta En[h_k]_a - En[h_k]_r \tag{13}$$

$$\Delta u_{jk} = \varphi u_{jk} + \beta En[v_j h_k]_a - En[v_j h_k]_r \tag{14}$$

In the last phase, the ODBN undergoes layer-by-layer training, using the initial variables established during the pre-training stage. This training is accomplished through a backpropagation model, which enhances the efficiency of variable adjustment. The objective of this fine-tuning process is to reduce the error, taking into account the results obtained from an extra layer on the above ODBN after every RBM training. The error term is described as:

$$Error = \frac{1}{2}\sum_{l=1}(y_l - h_l) \tag{15}$$

where *yl* and *hl* are the actual and reconstructed values of the *l^th^* node.

To achieve better performance from the DBN, this work presents an SCA for optimizing the hyperparameters of the approach. This optimizer aims at minimizing error terms to predict the test set with a high MSE value. Here, the hyper-parameters of the DBN like learning rate, size of batch, and epochs are optimized by the SCA. This optimizer influences the mathematical behaviour of the properties of sinh and cosh. The stages like Initialization, Exploration, Exploitation and Switch model are performed in the SCA.

Initialization: Similarly to various other metaheuristic optimizers, the SCA begins by initializing a set of candidate solutions Z in a random manner. Candidate solutions, as expressed in Equation (16), represent the initial state, with the best solution achieved throughout iterations being considered almost optimal.

$$Z = \begin{bmatrix} Z_{1,1} & Z_{1,i} \cdots & Z_{1,m} \\ Z_{2,1} & Z_{2,i} \cdots & Z_{2,m} \\ \cdots & \cdots\ \cdots & \cdots \\ Z_{M-1,1} & Z_{M-1,j} \cdots & Z_{M-1,m} \\ Z_{M,1} & Z_{M,2} \cdots & Z_{M,m} \end{bmatrix} \tag{16}$$

where *M* is the total number of candidate and the term *Z* is given as:

$$Z = r(M,\dim)(ul - ll) + ll \tag{17}$$

where *r, dim, ul* and *ll* are the random number, dimension, upper and lower limits.

Exploration: The optimization process divides exploration into two phases between iterations, and the presence of exploration in the later iterations is crucial to overcome local optima. The decision to switch between these phases is estimated by the value specified in Equation (18).

$$T = f\left(\frac{Max\_iter}{C}\right) \tag{18}$$

where *f* is the floor function, *Max_iter* and C are the maximum iteration and constant term. The position update for the initial phase is given as:

$$Z_{(i,j)}^{t+1} = \begin{cases} Z_b^{(j)} + r1 \times W1 \times Z_{(i,j)}^t & r2 > 0.5 \\ Z_b^{(j)} - r1 \times W1 \times Z_{(i,j)}^t & r2 < 0.5 \end{cases} \tag{19}$$

where $Z_{(i,j)}^{t+1}$ and $Z_{(i,j)}^{t}$ are the present and further iterations, $Z_b^{(j)}$ and *W*1 are the best position and weight coefficient, *r*1 and r2 are the random numbers. The term W_1 is computed by:

$$W1 = r3 \times a1 \times (\cosh r4 + v \times \sinh r4 - 1) \tag{20}$$

where *a*1 is the decreasing term, r^3 and r_4 are the random numbers. The term *a*1 is computed by:

$$a1 = 2\left(-1.3\frac{t}{Max\_iter} + n\right) \tag{21}$$

where *n* is the sensitive term.

During the next stage of exploration, the search agent exhibits minimal influence from the better solution attained, leading them to explore the next position nondirectionality for the present position. The calculation for updating the position is determined by the following:

$$Z_{(i,j)}^{t+1} = \begin{cases} Z_b^{(j)} + \left|\alpha \times W2 - Z_{(i,j)}^t\right| & r5 > 0.5 \\ Z_b^{(j)} - \left|\alpha \times W2 - Z_{(i,j)}^t\right| & r5 < 0.5 \end{cases} \tag{22}$$

where *α* is the small number and *W*2 is the weight coefficient. The term *W*2 is computed by:

$$W2 = r6 \times a2 \tag{23}$$

where *a*2 is the decreasing term and r^6 is the random number. The term *a*2 is computed by:

$$a2 = 2\left(-\frac{t}{Max\_iter} + n\right) \tag{24}$$

Exploitation: To maximize search space utilization, the exploitation process is split into two stages and occurs during all iterations. In the initial exploitation stage, emphasis is placed on exploiting the proximate space of *Z*, and the exploitation is formulated as:

$$Z_{(i,j)}^{t+1} = \begin{cases} Z_b^{(j)} + r7 \times W3 \times Z_{(i,j)}^t & r8 > 0.5 \\ Z_b^{(j)} - r7 \times W3 \times Z_{(i,j)}^t & r8 < 0.5 \end{cases} \tag{25}$$

where *r*7 and *r*8 are the random numbers and *W*3 is the weight coefficient and it is given as:

$$W3 = r9 \times a1 \times (\cosh r10 + v \times \sinh r10) \tag{26}$$

where *r*9 and *r*10 are the random numbers. During the next stage of exploitation, the candidate solutions engage in a thorough exploitation around the current optimized solution. The level of exploitation of the better solution attained intensifies with each subsequent iteration.

$$Z_{(i,j)}^{t+1} = Z_{(i,j)}^{t} + r11 \times \frac{\sinh r12}{\cosh r12}\left|W2 \times Z_b^{(j)} - Z_{(i,j)}^t\right| \tag{27}$$

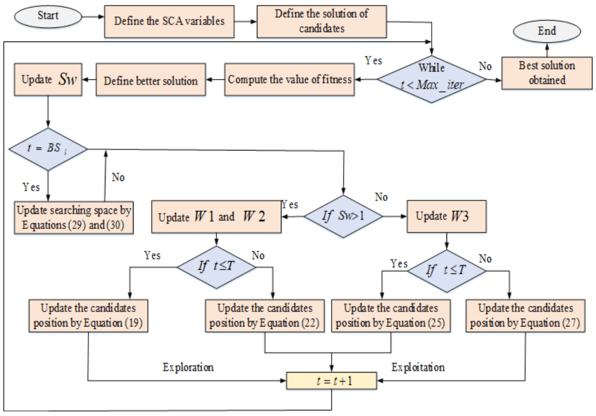

Start
Define the SCA variables
Define the solution of candidates
End
Yes
While t < Max_iter
No
Best solution obtained
Update Sw
Define better solution
Compute the value of fitness
t = BS_l
Yes
No
Update searching space by Equations (29) and (30)
Update W1 and W2
Yes
If Sw>1
No
Update W3
Yes
If t≤T
No
Yes
If t≤T
No
Update the candidates position by Equation (19)
Update the candidates position by Equation (22)
Update the candidates position by Equation (25)
Update the candidates position by Equation (27)
Exploration
t = t+1
Exploitation

Figure 4. SCA flow diagram

Bounded Search Model: To satisfy the complete search space, see Figure 4. SCA incorporates a model similar to the hunting of animals during the further phase and it is called as known as the Bounded searching model. This involves identifying the searching space by overall exploration of the searching region in the initial iteration.

$$BS_{l+1} = BS_l + f\left(\frac{Max\_iter - BS_l}{\beta}\right) \tag{28}$$

where BS_{l+1} is the total iterations that begin the present and further bounded searching, β is the sensitive term. If SCA utilizes the bounded searching model for all time, then the lower ll_l and upper ul_l limits are computed as:

$$ll_l = Z_b^{(j)} - \left(1 - \frac{t}{Max\_iter} + n\right) \times \left|Z_b^{(j)} - Z_s^{(j)}\right| \tag{29}$$

$$ul_l = Z_b^{(j)} + \left(1 - \frac{t}{Max\_iter} + n\right) \times \left|Z_b^{(j)} - Z_s^{(j)}\right| \tag{30}$$

where $Z_s^{(j)}$ is the sub-optimized term.

Switch model: In the SCA framework, a switch model incorporates sinh and cosh is introduced to alternate between exploration and exploitation. The primary emphasis of the switch model is on exploration during subsequent iterations to navigate the entire search space and overcome local optima. The expression to compute the switch model is given as:

$$Sw = \left(u - v\left(\frac{t}{Max\_iter}\right)^{\left(\frac{\cosh\frac{t}{Max\_iter}}{\sinh\frac{t}{Max\_iter}}\right)}\right) \times r13 \tag{31}$$

where *r*13 is the random number, *u* and *v* is the balancing term Figure 2 shows the SCA flow chart.

4. Results analysis

In this section, we delineate the experimental setup conducted on the Python platform, along with the description of the datasets exploited. Subsequently, a comparative analysis is carried out to show the efficacy of the suggested ODBN network against several existing models. Table 1 delineates the hyperparameters of the suggested ODBN network.

Table 1. Hyper-parameters

Hyper-parameters	Values
Learning rate	0.0001
Epochs	100
Number of HL	18
Size of batch	32, 64
Iterations	100
Population size	50

4.1 Dataset detail

The dataset considered is Light Detection and Ranging (LiDAR), which has parameters like aspect, altitude, Topographic Roughness Index (TRI), slope, Sediment Transport Index (STI), curvature, Stream Power Index (SPI) and Topographic Wetness Index (TWI). There are 144 flood events from 1920 to 2023.

4.2 Performance measures

The assessment metric for regression approaches is used to gauge the effectiveness of the model in predicting the output values based on the given input data. The metrics such as mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), root MSE (RMSE) and R-squared (R2) are measured to analyze the performance of the flood.

MAE: This metric acts as an alternative measure to analyze the differentiation among Z_l and $\hat{Z}_l$. Its calculation entails finding the absolute value of the differentiation among z_l and $\hat{Z}_l$ and subsequently means the absolute differentiation.

$$MAE = \frac{1}{p}\sum_{l}^{p}\left|z_l - \hat{z}_l\right| \tag{32}$$

MAPE: This metric, expressed as a percentage, characterizes the model's performance. To calculate it, the absolute value of the differentiation among Z_l and $\hat{Z}_l$, divided by Z_l, and the resulting percentages are then averaged.

$$MAPE = \frac{100}{p}\sum_{l}^{p}\frac{z_l - \hat{z}_l}{z_l} \tag{33}$$

MSE: This measure is the differentiation of predicted $Z_l$and actual values $\hat{Z}_l$ and the squared value of MSE is the RMSE.

$$MSE = \frac{1}{p}\sum_{l}^{p}\left(z_l - \hat{z}_l\right)^2 \tag{34}$$

$$RMSE = \sqrt{\frac{1}{p}\sum_{l}^{p}\left(z_l - \hat{z}_l\right)^2} \tag{35}$$

R2: It serves as an indicator of how well a model is set up with the data set. To calculate it, the sum of the squares of the differentiation among $Z_l$and $\hat{Z}_l$. is determined, and this total is then divided by the sum of the squares of differentiation among the mean of Z_l and $\vec{z}_l$.

$$R^2 = 1 - \frac{\sum\left(z_l - \hat{z}_l\right)^2}{\sum\left(z_l - \vec{z}_l\right)^2} \tag{36}$$

Cohen's kappa index: It is the proportion of observed value A_o over predicted value A_p and it is defined as:

$$kappa = \frac{A_o - A_p}{1 - A_p} \tag{37}$$

4.3 Comparative analysis

In this section, an examination and comparison of the evaluation metrics for the proposed ODBN approach are conducted in comparison to several DL approaches like RNN, LSTM and DBN. Evaluation represents a crucial stage in model evaluation, providing valuable insights to identify the optimal approach based on performance results.

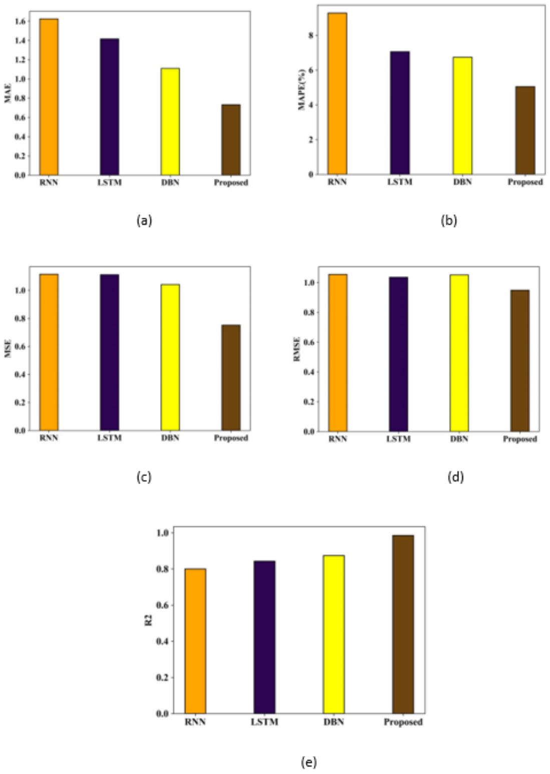


Figure 4. Comparison of (a) MAE, (b) MAPE, (c) MSE, (d) RMSE and (e) R2

Figure 4 and Table 3 show the comparison of metrics such as MAE, MAPE, MSE, RMSE and R2. It is noted that MAE values achieved by RNN, LSTM, DBN and the proposed ODBN are 1.62, 1.41, 1.10 and 0.732. Then, the MAPE values achieved by the RNN, LSTM, DBN, and the proposed ODBN are 9.27, 7.05, 6.73 and 5.047. Similarly, for the better flood forecasting model, the MSE and RMSE values must be very low, and the proposed ODBN achieved the better values of 0.752 (MSE) and 0.948 (RMSE). The value of the R2 values must be very high, and it is noted that the proposed ODBN achieved a better R2 value of 0.985.

Table 2. Comparison of the different approaches

Methods	RNN	LSTM	DBN	Proposed (ODBN)
MAE	1.62	1.41	1.10	0.732
MAPE	9.27	7.05	6.73	5.047
MSE	1.11	1.11	1.04	0.752
RMSE	1.05	1.03	1.05	0.948
R2	0.80	0.84	0.87	0.985

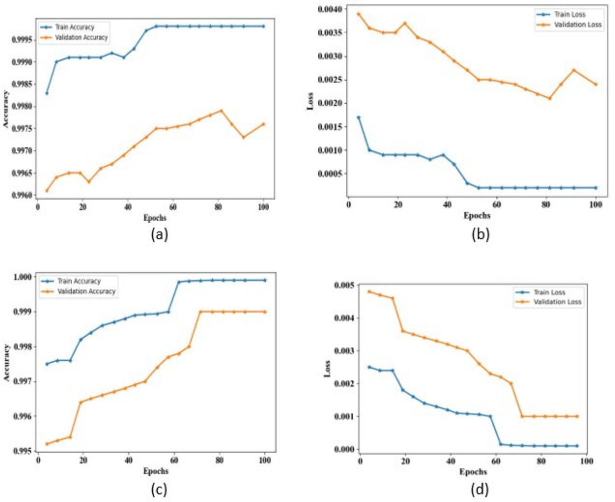


Figure 4. Accuracy and loss by varying the size of batch (a) 32 and (b) 64

Figure 4 depicts the accuracy and loss by varying the sizes of batches of 32 and 64 respectively. It is observed that the accuracies increase after the 60th epoch of training. Similarly, it is observed that losses decrease after the 60th epoch of training. Graphs are generated to visualize the train and validation values, revealing that the model does not exhibit under- or over-fitting. The model demonstrates superior generalization, which confirms the efficacy of the suggested ODBN in the flood forecasting process.

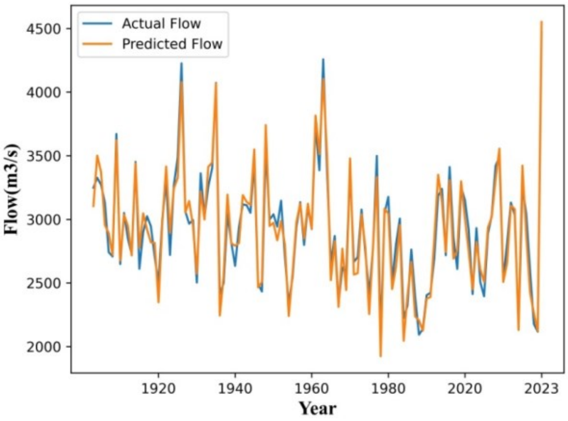


Figure 5. Water flow prediction of the proposed ODBN

Figure 5 shows the water flow prediction of the suggested ODBN model. The analysis is carried out between the years 1920 and 2023. When the performance of the flood is forecasted, the predicted flood and the actual flood are compared. In particular, the predicted flow generated by the suggested ODBN corresponds correctly to the actual flood values.

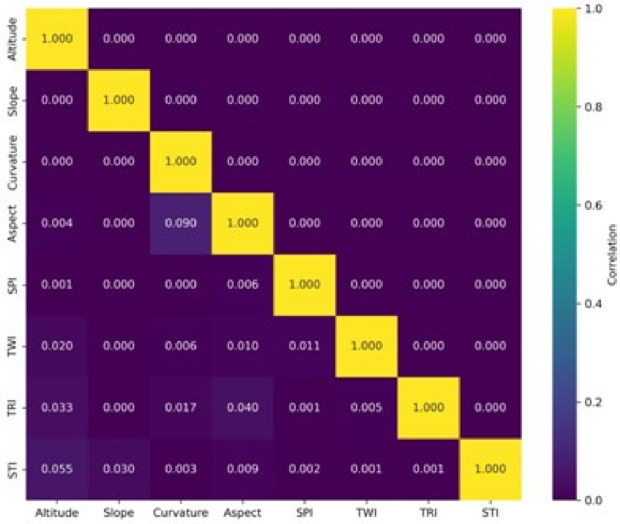


Figure 6. PC of the proposed ODBN

PC (Pearson correlation) is visualized in a square table, and this matrix provides the calculated PC for every set of columns within a set of parameters. Figure 6 shows the Pearson coefficient of the proposed ODBN considering factors such as aspect, altitude, TRI, STI, curvature, slope, SPI and TWI. It is observed that the Pearson coefficient value is almost 1 for all the factors.

Table 4. Analysis of Cohen’s Kappa Index

Factors	Cohen’s kappa index
Not considering aspect	89
Not considering altitude	71
Not considering TRI	77
Not considering STI	89
Not considering curvature	75
Not considering slope	74
Not considering SPI	83
Not considering TWI	83

Table 4 presents the analysis of Cohen’s kappa index and the results reveal that the highest precision is observed when altitude and slope are not considered. This suggests that these factors have the greatest influence on flood occurrence.

4. Conclusions

This work presented an ODBN model for flood foresting considering different parameters. Initially, the data set was preprocessed by linear interpolation. Then the ODBN

DL model was presented for automated feature extraction and flood forecasting. Here, the hyper-parameters of the standard DBN were optimized by the SCA. The experimental analysis was carried out on the real-time dataset and achieved better MAE and R2 values of 0.732 and 0.985 respectively. The findings suggest that the prediction performance of the ODBN model was improved. In addition, this ODBN model gives precedence to essential data, providing more precise flood predictions. They adeptly capture intricate features, positioning them as potential leaders in advancing flood forecasting systems. In future, the proposed work need to test in different cities with different paramenter.

References

Bentivoglio R., Isufi E., Jonkman S.N. and Taormina R. (2022). Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrology and Earth System Sciences* **26**, no. 16 (2022): 4345–4378.

Chen C., Jiang J., Liao Z., Zhou Y., Wang H. and Pei Q. (2022). A short-term flood prediction based on spatial deep learning network: A case study for Xi County, China. *Journal of Hydrology* **607**, 127535.

Chenmin N., Marsani M.F. and Pei Shan F. (2024). Flood prediction based on feature selection and a hybrid deep learning network. *Journal of Water and Climate Change* , jwc2024559.

Das J., Manikanta V., Nikhil Teja K. and Umamahesh N.V., (2022). Two decades of ensemble flood forecasting: A state-of-the-art on past developments, present applications and future opportunities. *Hydrological Sciences Journal* **67**, no. 3, 477–493.

Ebtehaj I. and Bonakdari H. (2022). A reliable hybrid outlier robust non-tuned rapid machine learning model for multi-step ahead flood forecasting in Quebec, Canada. *Journal of Hydrology* **614,** 128592.

Farahmand H., Xu Y. and Mostafavi A. (2023). A spatial–temporal graph deep learning model for urban flood nowcasting leveraging heterogeneous community features. *Scientific Reports* 13, no. 1, 6768.

Hakim D.K., Gernowo R. and Nirwansyah A.W. (2023). Flood prediction with time series data mining: Systematic review. *Natural Hazards Research*.

Han S. and Coulibaly P. (2017). Bayesian flood forecasting methods: A review. *Journal of Hydrology* **551,** 340–351.

Hosseiny H. (2021). A deep learning model for predicting river flood depth and extent. *Environmental Modelling & Software* **145**, 105186.

Hu R., Fang F., Pain C.C. and Navon I.M. (2019). Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method. *Journal of Hydrology*, **575**, 911–920.

Jain S.K., Mani P., Jain S.K., Prakash P., Singh V.P., Tullos D., Kumar S., Agarwal S.P. and Dimri A.P. (2018). A Brief review of flood forecasting techniques and their applications. *International Journal of River Basin Management* 16, no. 3

Lei X., Chen W., Panahi M., Falah F., Rahmati O., Uuemaa E., Kalantari Z. *et al*. (2021). Urban flood modeling using deep-learning approaches in Seoul, South Korea. *Journal of Hydrology* 601: 126684.

Linh N.T.T., Ruigar H., Golian S., Bawoke G.T., Gupta V., Rahman K.U., Sankaran A. and Pham Q.B. (2021). Flood prediction based on climatic signals using wavelet neural network. *Acta Geophysica* **69**, no. 4, 1413–1426.

Löwe R., Böhm J., Jensen D.G., Leandro J. and Rasmussen S.H. (2021). U-FLOOD–Topographic deep learning for predicting urban pluvial flood water depth. *Journal of Hydrology* **603**, 126898.

Luppichini M., Barsanti M., Giannecchini R. and Bini M. (2022). Deep learning models to predict flood events in fast-flowing watersheds. *Science of The Total Environment* **813,** 151885.

Nevo S., Morin E., Gerzi Rosenthal A., Metzger A., Barshai C., Weitzner D., Voloshin D. *et al*. (2022). Flood forecasting with machine learning models in an operational framework. *Hydrology and Earth System Sciences* 26, no. 15, 4013–4032.

Onen F. and Bagatur T. (2017). Prediction of flood frequency factor for Gumbel distribution using regression and GEP model. *Arabian Journal for Science and Engineering* **42** (2017): 3895–3906.

Panahi M., Jaafari A., Shirzadi A., Shahabi H., Rahmati O., Omidvar E., Lee S. and Bui D.T. (2021). Deep learning neural networks for spatially explicit prediction of flash flood probability. *Geoscience Frontiers* **12**, no. 3, 101076.

Piadeh F., Behzadian K. and Alani A.M. (2022). A critical review of real-time modelling of flood forecasting in urban drainage systems. *Journal of Hydrology* **607,** 127476.329–344.

Santhanaraj R.K., Rajendran S., Romero C.A.T. and Murugaraj S.S. (2023). Internet of Things Enabled Energy Aware Metaheuristic Clustering for Real -Time Disaster Management. *Computer Systems Science and Engineering*., **45**, 1561–1576.

Surendran R., Alotaibi Y. and Subahi A.F. (2023). Lens-Oppositional Wild Geese Optimization Based Clustering Scheme for Wireless Sensor Networks Assists Real Time Disaster Management. *Computer Systems Science and Engineering*., **46**(1), 835–851.

Surendran R., Alotaibi Y. and Subahi A.F. (2023). Wind Speed Prediction Using Chicken Swarm Optimization with Deep Learning Model. *Computer Systems Science & Engineering*, **46**, 3.

Surendran R., Tamilvizhi T. and Lakshmi S. (2021). Integrating the Meteorological Data into a Smart City Service Using Cloud of Things (CoT). In Emerging Technologies in Computing: 4th EAI/IAER International Conference, iCETiC 2021, Virtual Event, August 18–19, *Springer International Publishing*, 4, 94–111.

Zhong M., Zhang H., Jiang T., Guo J., Zhu J., Wang D. and Chen X., (2023). A hybrid model combining the cama-flood model and deep learning methods for streamflow prediction. *Water Resources Management* **37**, no. 12, 4841–4859.

Zou Y., Wang J., Lei P. and Li Y. (2023). A novel multi-step ahead forecasting model for flood based on time residual LSTM. *Journal of Hydrology* **620**, 129521.