Flood Prediction in Chennai based on Extended Elman Spiking Neural Network using a Robust Chaotic Artificial Hummingbird optimizer

Karthik .S[1], Surendran .R[2], Sam Kumar G.V[3], Senduru Srinivasulu[4]

[1] Department of Computer Science and Business Systems,
Muthayammal Engineering College (Autonomous), Rasipuram, 637408, Tamil Nadu, India.
karthiks1087@gmail.com

[2]Department of Computer Science and Engineering, Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences, Chennai, 602105, India.
dr.surendran.cse@gmail.com

[3] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, 522302, Andhra Pradesh, India.
gvsamkumar@gmail.com

[4] School of Computing, Department of Computer Science and Engineering, Sathyabama Institute of science and Technology, Chennai, 600119, India.
sendurusrinivasulu.cse@sathyabama.ac

*Corresponding author: E-mail: dr.surendran.cse@gmail.com

ABSTRACT

The Chennai region's meteorological conditions are causing floods in many districts to occur more frequently and with greater intensity. Therefore, anticipating and planning for floods under severe weather conditions is essential for making decisions and handling impending calamities. These days, deep learning (DL) methods are crucial for supporting meteorological applications and successfully preventing natural hazards. The use of climatological data to enhance flood prediction is covered in this manuscript. By examining the effects of rainfall changes in several Chennai regions, the current study attempts to provide an accurate estimate of flood risks. Pre-processing and flood forecasting are the two stages that are completed for

the proposed framework. The MaxAbsScaler (MAS) approach, which could be used to eliminate unwanted missing values from the database, is described in the preparation step. In order to predict the flood in various Chennai regions, the Extended Elman Spiking Neural Network (ExESNN) technique is then suggested. In order to avoid network problems during the training phase, the derived model's parameters are tuned using the Chaotic Artificial Hummingbird Optimizer (Ch-AHO). The Python platform is utilized to process the built framework, and the experimentation process makes use of the real-time flood prediction database gathered between 2000 and 2023. Numerous computational metrics are assessed and differentiated from various research, including R2, mean absolute error (MAE), Kling-Gupta Efficiency (KGE), Nash-Sutcliffe Efficiency (NSE), root mean square error (RMSE), and calculation time. In comparison to several traditional research on rainfall forecasting in various parts of Chennai, the developed technique yields an overall R2 of 0.994, RMSE of 0.851, KGE of 0.968, NSE of 0.991, MAE of 0.50, and overall CT of 63.33s.

Keywords: Flood Prediction, Deep Learning, Extended Elman Spiking Neural Network, Chaotic Artificial Hummingbird Optimizer.

1. Introduction

Flood prediction is an important study in disaster management, more so because of the high intensities and increased frequencies of climatic events. Dtissibe et al (2024) found amongst the worst disasters that can strike a community include floods, which result in massive losses in lives and population displacements alongside damage to infrastructure. Surendran et al. (2023) said wind speed based one of the highly susceptible regions in India, has been heavily hit by floods in 2015. Zhong et al (2023) and Adnan et al. (2023) across the ancient era, Chennai has seen an upsurge in the number of generous rain days, unresolved the thresholds set at 95% and 99% of the typical precipitation from 1910 to 2022. Conventional methods in flood

forecasting widely used by hydrologists largely bank on manual methods, historical hydrological information analysis, and simple statistical devices. Alotaibi Y et al (2024) introduced these general methods are the basic foundation in flood prediction but to come packed with some striking limitations. Rahman et al. (2023) implements the inaccuracy residue on the use of manual techniques is inevitable since they fail to capture complex, often non-linear relationships between meteorological variables and flood events. Ouma et al. (2023) executed largely depend on analogues incapable of fully representing unprecedented conditions of meteorological extremes and changes in land use.

More to that, such techniques have shortcomings related to the quality and quantity of data. In the cases of sparse or otherwise irregular data, reliable predictions are unlikely to be generated. He et al. (2023) and Lin et al. (2023) developed a manual perusal of voluminous data is quite tiring; thus, delays could most likely lead to an inefficient flood warning system apart from human errors. In addition, manual models tend to be static and, therefore, not able to adapt dynamically to new input data presented with changing environmental circumstances. Xu et al. (2023) and Nguyen et al. (2023) established the addition of machine learning has brought in data-driven flood prediction. Alotaibi Y et al. (2024) provide a ML models work in a way that the algorithms detect a pattern and make forecasts based on huge piles of data. In this regard, although machine learning models improve the results with respect to traditional methods, there lie several challenges one faces. Farahmand et al. (2023) and Zhong et al. (2024) handled the quality of the predictions made by the ML model mostly depends on the quality of the data with which it has been trained. Incomplete, noisy, or biased data may undermine the performance of the model. Aydin et al. (2023) added some ML models are at large risk of overfitting, such that a model might perform well on training data but poorly on new data; this tends to make flood forecasts less reliable. Dehghani et al. (2023) compiled many ML models

are black boxes, and their decision-making processes are generally unclear. Such non-transparency hampers practical applications and acceptance.

Wu et al. (2023) and Lin et al. (2024) established advanced ML models also place a high demand on computation resources and expertise, which remains out of reach for many organizations, especially in resource-limited settings. Shikhteymour et al. (2023 implements a Deep learning (DL), a subset of machine learning (ML), has emerged as a promising solution for such challenges. DL, using multi-layer neural networks, captures complex non-linear data patterns for learning and generalization from large volumes of diversified sources. Surendran R et al. (2023) executed this is what makes it very effective to actually make flood predictions, since it can analyze the many inputs coming from satellite imagery, weather forecasts, and hydrological data in much more detail. DL therefore provides the large jump necessary to improve accuracy while increasing the adaptability of flood forecasting by several folds, and considered quite significant. It comes as a major advancement in this field, as it provides much better precision and responsiveness in flood risk management. DL heralds an exciting new phase in the evolution of flood prediction, with promise for more effective and efficient disaster preparedness and response.

Motivation**:** Chennai, being one of the southernmost region of India, stands a particular chance of being inundated as a result of climatic conditions. Indeed, the region is subject to furious rains during the Great Floods, which may result in catastrophic floods, especially when the rainfall is heavy or there are extreme weather events. Such floods happen and affect millions of dwellers, destroying infrastructure and property, with many lives being lost. In the past years, Chennai has been hit by some really bad floods—2015 and 2019 floods that hit, after which the need was felt more than any time before to have better flood prediction and management systems. This need becomes apparent against the kind of traditional methods available for flood prediction used in Chennai—manual hydrological models and basic

statistical approaches—these methods will rarely stand up against the increasing frequency and intensity of flood events. These methods normally rely on historical information and rudimentary analyses, perhaps not necessarily reflecting the complexity of the meteorological and hydrological dynamics in the region. The shortcomings of these ideas are in accuracy, sluggishness of response, and incapability to manage changes in conditions that become rapidly changing. DL provides transformations for these challenges. DL models, built with the help of advanced neural network architectures, have the capability to process voluminous data coming in from different sources like satellite imagery, weather predictions, or real-time hydrological data.

The main aim of the developed framework is to propose an innovative optimized deep learning (DL) model for forecasting the rainfall across various regions in India.

The Objectives are encompassed as follows:

- To introduce a MaxAbsScaler (MAS) technique to eliminate unwanted missing values present in the flood data.
- To elucidate an extended Elman spiking neural network (ExESNN) technique to forecast the flood that occurs in various regions in Chennai state.

The scope of the proposed work in to test the performance of the developed model with different evaluation measures of performance assessment, including R2, RMSE, Kling-Gupta Efficiency (KGE), Nash-Sutcliffe Efficiency (NGE), mean absolute error (MAE), and computation time, with reference to various conventional studies.

2. Related Works

Moon et al. (2023) established the prediction of urban floods using combined DL models. In this study, runoff from the precipitation model with a DL model was introduced to create an integrated modeling technique that served as the foundation for a flood forecasting model. Additionally, a technique was emphasized that makes use of many representative rainfall

characteristics for determining the likelihood of flooding. The research concentrated on metropolitan streams, overall precipitation totals, time span, and Time-dependent distribution to generate precipitation situations virtually. Furthermore, the runoff from the precipitation model's projected findings was utilized as input information to predict when floods occur during periods of heavy rain. With a strong association, the anticipated results showed a high accuracy, NSE and correlation. However, this method reduces the interpretability while processing new or complex time-series data.

Zhou et al. (2023) defined the DL-based time-series data model for determining the flood conditions using spatial and temporal attributes. To create a DL-based information-driven model for forecasting floods in both temporal and geographical aspects. The model's effectiveness was tested using an actual scenario in northern China. The findings unambiguously show that the model can forecast elevated water elevations and overflow intervals for a variety of hyetograph sources with significant energy savings. The model outperformed the real-world-based hydrological model in the prediction of flood maps, 19 585 times better. However, the long-term dependency problems were high while training with larger data.

Wang et al. (2023) presented the rapid urban overflow forecasts based on clustering and optimized DL model. In this work, a KM cluster and Bayesian optimal DL algorithm-based metropolitan flooding forecast framework were emphasized. Based on variables associated with a disaster-bearing setting, the examined area was divided into several clusters related to flooding causes using the KM cluster process. To create an estimate model for the locations experiencing identical inundation causes, the gradient boosting DT (GBDT) technique was used. Additionally, the study looked into using the Bayesian approach to adjust the critical parameters of the GBDT model to increase model accuracy. China's Haikou region served as the model's verification site. The study region was divided into three clusters utilizing the KM

cluster based on the following disaster-bearing environment factors: altitude, gradient, tube infrastructure thickness, proximity to the stream, and bending. However, the gradient insufficiency problems were high while training with larger data.

Windheuser, L., R et al. (2023) put forth the DL based end-end model for determining the flood forecasts effectively. In this study, a completely computerized throughout its entirety image detection procedure was considered to use DNN for forecasting flood level statistics between two USGS metering points in Georgia, USA: Columbus and Sweetwater Creek.The USGS live river web cams, which were placed thoughtfully close to the observation points and were updated approximately every 30 seconds, provided the photographs. In order to reduce the amount of photos required for training, a U-Net-CNN (U-Net CNN) was initially layered on top of a segmentation procedure for distortion and feature elimination to determine the overwhelming stage. The catastrophe stage historical data was subsequently trained to be predicted in almost real-time using an LSTM, a CNN, and an intensive model. However, the error was high when larger database was utilized for the execution process.

Zhang et al. (2023) pointed out the effective temporal urban flood forecasting based on combined DL model. This paper suggested an attention-based LSTM-DW with a weighted MSE (WMSE) loss function and double-time sliding windows (DTSW). In Shenzhen, China, three flood-prone metro areas were subjected to the ALSTM-DW paradigm and several comparison tests were used to confirm its efficacy. The findings suggest that the model under consideration performs well with a maximum flow error smaller than 0.85 and a coefficient of determination bigger than 0.85, less than 0.015 m, with an average duration to peak inaccuracy during testing of less than 2 min, and reduce excessive swings in hydrograph forecasts brought on by a disproportionate response to precipitation. However, attention mechanisms can introduce additional hyper parameters that need to be tuned, potentially leading to increased complexity and overfitting.

Surendran R., et al. (2021) introduced the interpretable integrated DL model for determine the flood conditions using real-time data. Here, AGRS-LSTM-Transformer, an interpretable combination model was utilized for anticipating flooding that combines the strengths of the Transformer, LSTM, and Adaptive Random Search Algorithm (AGRS). By using flood statistics from the Jingle watershed from 1971 to 2013, this study examines the predicted efficacy of the combined approach and evaluates it against other schemes. However, the utilized transformer model maximizes the time consumption and labor-costs.

Weng et al. (2023) determined the model for forecasting the flood susceptibility using ensemble techniques. The study involves selecting a case study, in Dingnan County in the Jiangxi Province of China. Single flood event points and random sampling methods were employed to produce flood and non-flood data. Next, the frequency proportion was used to evaluate the connection between every factor that impacts the probability of floods. The results indicate that there is no convergence between the impacting factors; a possible ten of them have contributed to the event of floods, and all of them are used to create the connection model. Finally, vulnerability to flooding maps were created using the DL, FC-DL, RF-DL, and RSS-DL. However, these models were computationally expensive and resource-intensive

Babu T et al. (2024) defined the time series DL model for automatic flood forecasting. Two of the most recent generative adversarial network (GAN) variations, TimeGAN and Real-world RTSGAN, were used to create synthetic flood time series while taking into account the rarity of flooding incidents and the substantial dimension of overflow time series. Results demonstrate that time-series GANs may effectively and reliably simulate the topographical relationships between flood series from various locations in an instance assessment of the Xijiang River Basin in southern China. RTSGAN surpasses TimeGAN, particularly when the intervals of periods are long. In addition, realistic patterns and a further number of simulated segments were used to train the flood projection algorithms for GBRT, LSTM, and QRLSTM.

However, GAN models can be sensitive to hyper parameters and training data, leading to variability in performance.

2.1. Problem Statement

Geographically, Chennai lies in a region highly prone to severe floods, coupled with intense monsoon rainfall and complicated river systems. It has witnessed severe flood events in the recent past, causing vast infrastructure damage, disruption of life, and massive economic loss. Traditional methods of flood prediction using manual hydrological models and simple statistical techniques are inadequate to meet up with the challenges in tackling such extreme weather events. Most of these methods are normally restricted in terms of accuracy and response time and can barely capture the dynamic and nonlinear relationships between meteorological conditions and flood occurrences. In fact, the crux of the matter is none other than that the existing flood prediction system in Chennai cannot provide timely, accurate, and reliable forecasts for effectively supporting flood management and disaster preparedness. Currently, methods are based on historical data and simplistic models that might be incapable of fully capturing the complexity of real-time weather patterns, land use changes, and hydrological conditions. Ultimately, this flood prediction would not be very accurate and thus, in general, it leads to poor preparation and response, which places the community at greater risk, adding to the cost in economic and human terms.

3. Developed Methodology

In this paper, the application of climatological data in the improvement of flood prediction is discussed. The present work aims at making an exact estimation of flood risks by studying the impact of rainfall variations in different regions of Chennai. The developed framework undergoes two phases: pre-processing and flood forecasting. The preprocessing stage outlines the MaxAbsScaler (MAS) technique that could be used to remove unsolicited missing values from the database. Afterwards, the Extended Elman Spiking Neural Network (ExESNN) technique is proposed in order to envisage the flood in different regions of Chennai. Figure 1 indicates the flow of the developed Framework

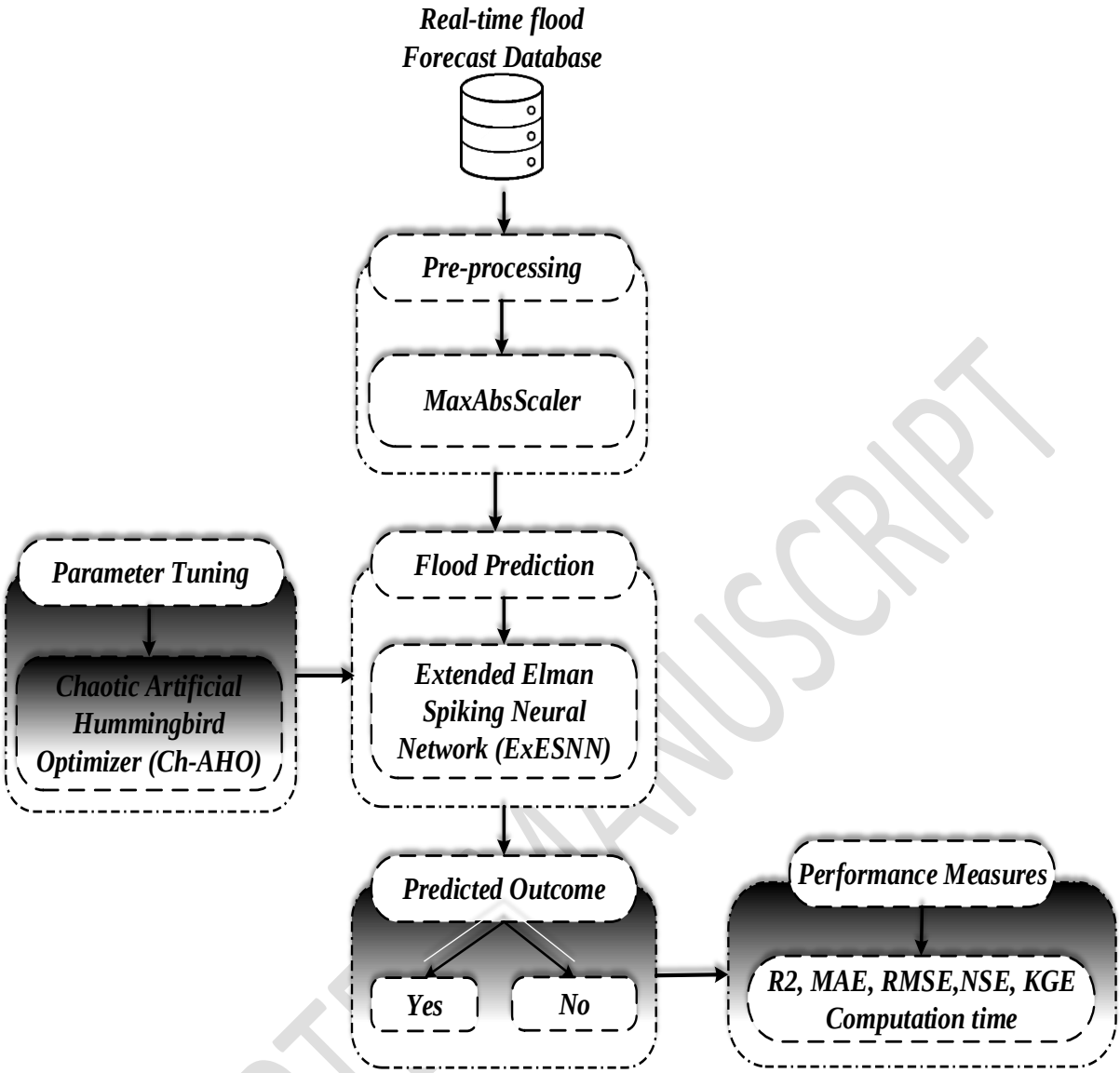

Figure 1. The flow of the developed Framework

3.1. Preprocessing Stage

The raw data is initially fed into the Max Abs-Scaler normalization (MAS) technique to remove irrelevant and discontinuous data points. Arun Mozhi Selvi Sundarapandi, at al. (2024) shared the given dataset contains high ambiguity and volatility, data normalization is performed using the MAS technique and it is scaled to the highest value ranging from -1 to 1. It separates every observation using the highest possible variables and it can be formulated in equation (1),

$$z_{scaled} = \frac{z}{\max(z)} \tag{1}$$

The out-dated transformation determines the disperses within the values ranges between -1 to 1.

3.2. Chennai Flood Prediction using ExESNN Technique

After pre-processing, flood forecasting are undertaken to minimize economic loss and attempt to reduce the exposure level of urbanized areas. Nowadays, the DL technique plays an integral role in predicting weather conditions with minimal time complexity. This study proposes a novel Extended Elman Spiking Neural Network (ExESNN) to forecast the flood conditions in various regions of Chennai. The proposed ExESNN technique is the extended form of the fundamental Elman network model, which is part of the recurrent spiking network model. The developed architecture contains four layers namely the input layer, the context layer, the hidden layer, and the outcome layer. The extended architecture contains self-feedback with differential gain in the context layer, in which the feedback from the hidden layer to the context layer has feedback weights, w_{hc} which are enhanced during the execution process. The spiking function increases the training process with the use of active nodes by updating the threshold values. The dynamicity of the ExESNN can be mathematically formulated as,

$$Z(x) = f\left(w_{zc} Z_c(x), w_{zu} V(x)\right) \tag{2}$$

$$Z_c(x) = \lambda(x) Z_c(x-1) + w_{hc} Z(x-1) \tag{3}$$

$$P_m(x+1) = w_{uz} Z(x) \tag{4}$$

Here, $P_m(x)$ and $V(x)$ indicates the input and output of ExESNN respectively. The parameter $Z_c(x)$ and $Z(x)$ deliberates the vector states of context and hidden layers respectively. The parameter w_{zu}, w_{zc}, and w_{px} signifies the vector weights between hidden and input layers, between context and hidden layers, and between hidden and outcome layers respectively. Moreover, $f(.)$ represents the nonlinear attribute which indicates the ExESNN performance. The self-feedback λ in the context layer is enhanced in the developed model it attains the accurate value. The internal neurons in the ExESNN technique indicate the single synaptic lethal between them. The neuron j is not permissible to spike via the remaining time interval

K when the threshold exceeds a specific sample k_j and recovers again in $k_j + c^x$. Every single link between the layers in ExESNN is comprised of a set with a similar number of synaptic linkages. Each sub-link is connected with varying weights and latency. The changes between the time of presynaptic neurons j and postsynaptic neurons are determined based on synaptic linkages. At this time, the postsynaptic neurons start to increase and cause synapse series in the links. Each synapse weight affects the spike function ψ indicating the neuron activation function. Figure 2 encompasses the Architecture of ExESNN Technique.

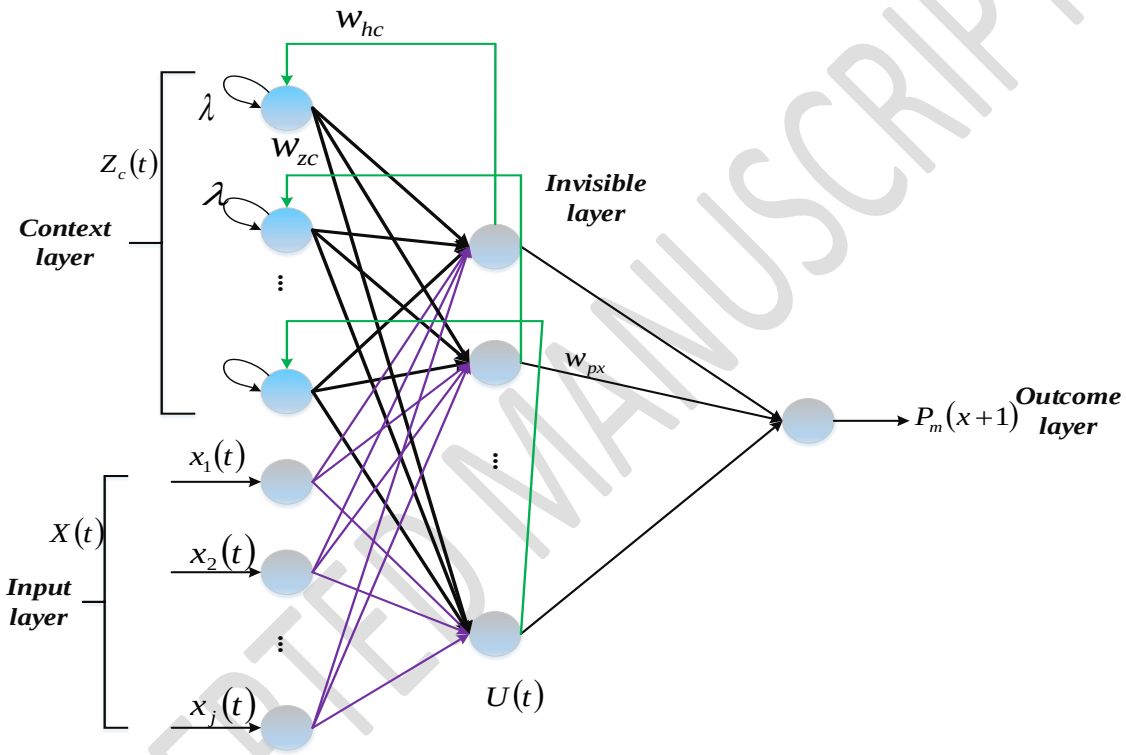

Figure 2. Architecture of ExESNN Technique

In the ExESNN technique, the training process is altered and weights are updated. The parameters $p_y(x)$, $R(x)$ indicates the outcome and reference data, while the parameters $E(x)$ and $\tilde{E}(x)$ indicates the error and differential error respectively. The parameters of the ExESNN technique are learned according to the time series data in a closed-loop format. The parameter r' indicates the error changes. The single-layered ExESNN has dual input neurons, a single output layer, six hidden layers, and a context layer. The error is considered the inputs and error changes with seven synapses for each link. The initial weights of the ExESNN technique are created arbitrarily within the range $[-0.5, 0.5]$ and learning rate and self-feedback are started at 0.01 and 0.5 respectively for training 100 epochs. At the time of input data

synthesis of the ExESNN technique, the error and the differential error are altered into different spikes.

The outcome data of ExESNN is also synthesized with the spike time, which is modified into an actual value as input. The proposed ExESNN works based on the negative gradient descent scheme to reduce the difference between the actual and forecasted outcomes. Various parameters like weights, sub-links, or synaptic latencies and thresholds. Combining the ENN with the SNN is chosen to determine the efficacy of the developed model. The hidden layers minimize the training speed and maximize the system complexities. The number of sub-links between input and hidden layers is updated along with the outcome and hidden layers. At the initial stage, weights are arbitrarily selected and iterated continuously with a particular learning rate.

The response is initially encoded into different spike times and it can be mathematically formulated as,

$$k_i^f = k_{\max} - \left\lfloor \frac{k_{\min}\left(Rf\left(k\right) - Rf_{\min}\right)\left(k_{\max} - k_{\min}\right)}{\left(Rf_{\max} - Rf_{\min}\right)} \right\rfloor \tag{5}$$

Here, $Rf_{\max}$ and $Rf_{\min}$ indicates the maximum and minimum actual response, $k_{\max}$ and $k_{\min}$ indicates the maximum and minimum interval time respectively. The decoding part of real response can be mathematically formulated as,

$$Rf\left(k_i\right) = \frac{\left(k_{\max} - k_i - k_{\min}\right)\left(Rf_{\max} - Rf_{\min}\right)}{\left(k_{\max} - k_{\min}\right)} + Rf_{\min} \tag{6}$$

There are totally two phases namely feed-forward (FF) phases in which each spike neurons at every time interval K only once and this arises when the threshold value is increased by the potential membrane η. The FF phase starts from the hidden layer and the neuron is trained continuously to verify whether it is spiked or not. The ExESNN technique utilizes the upcoming neurons when the existing neurons have been spiked. The potential membrane $n_j\left(k\right)$ is evaluated by the training mechanism based on the subsequent equation using the input spikes k_i^f at the input layer of the neuron can be mathematically formulated as,

$$n_j(k)=\sum_{i=1}^{nh}\sum_{t=1}^{del}W_{zu}^{t}\beta\left(k-k_i^f-q^t\right)+\varsigma\sum_{i=1}^{nh}\sum_{t=1}^{del}W_{zc}^{t}\times Rf_{hi}^{t}(k-1) \tag{7}$$

The parameter $Rf_{hi}^{t}(k-1)$ indicates the existing output from the hidden layer and the current input. The function $\beta\left(k-k_i^f-q^t\right)$ can be mathematically formulated as,

$$\beta\left(k-k_i^f-q^t\right)=-\sigma\times\exp\left(\frac{-\left(k-k_i^f-q^t\right)}{\gamma}\right) \tag{8}$$

The sub-link weights are updated when the FF mode is completed. In contradiction to FF, back-propagation starts from the outcome layer and returns to the hidden layer. The sub-link outcome is updated and it can be mathematically formulated as,

$$w_{pz}^{t}(k+1)=w_{pz}^{t}(k)-\Delta w_{pz}^{t}(k) \tag{9}$$

Here, $\Delta w_{pz}^{t}(k)=\mu\vartheta_i Z_t$. The error between the final spike time of the outcome neuron and its real firing time can be formulated as,

$$E=\left(K_i^q-k_i^f\right) \tag{10}$$

The parameter ϑ_i can be mathematically formulated as,

$$\vartheta_i=\frac{E}{\sum_{i=1}^{nh}\sum_{t=1}^{del}W_{zp}^{t}\frac{\partial Z_j^t}{\partial k}} \tag{11}$$

The sub-links of the hidden layer are updated using the mathematical formulation depicted below:

$$w_{zu}^{t}(k+1)=w_{zu}^{t}(k)-\Delta w_{zu}^{t}(k) \tag{12}$$

Here, $\Delta w_{zu}^{t}(k)=\mu\vartheta_j X^{t}{}_{j}$ whereas, ϑ_j can be mathematically formulated as,

$$\vartheta_j = \frac{\sum_{i=1}^{ni} \vartheta_j \sum_{t=1}^{del} w_{zu}^t \frac{\partial X^t}{\partial k}}{\sum_{t=1}^{del} w_{zu}^t \frac{\partial zc^t}{\partial k}} \tag{13}$$

The context layer weights are updated and it can be mathematically formulated as,

$$w_{zc}^t(k+1) = w_{zc}^t(k) - \Delta w_{zc}^t(k) \tag{14}$$

The updation of sub-link latency and neuron threshold can be mathematically formulated as,

$$\Delta_{del}^t = \zeta_d \sum_{j=1}^{NH} \frac{\partial E}{\partial k_i^f} \frac{\partial k_i^f}{\partial X_j(k)} \frac{\partial X_j(k)}{\partial del^t} \tag{15}$$

$$\Delta\omega = \zeta_\omega \sum_{j=1}^{NH} \frac{\partial E}{\partial k_i^f} \frac{\partial k_i^f}{\partial X_j(k)} \frac{\partial X_j(k)}{\partial \omega^t} \tag{16}$$

Here, n_j indicates the potential membrane of the neuron j, w_{pz}^t indicates the sub-link weights of the outcome layer, w_{pz}^t indicates the sub-link weights of the hidden layer, w_{zu}^t indicates the sub-link weights of the input layer, t indicates the step time, μ deliberates the learning rate, del indicates the sub-link-latencies, K_i^q indicates the target outcome of neuron spike time, k_i^f signifies the real outcome of neuron spike time, q^t denotes the connection delay, NH deliberates the total neurons in hidden layer, nh deliberates the total neurons in hidden layer, ni deliberates the total neurons in input layer, σ deliberates the constant ranges between 0 to 1, K indicates the time, ζ_d signifies the sub-link latency learning rate, ζ_θ represents the sub-link threshold learning rate, k encompasses the time duration, ω encompasses the threshold parameter, γ signifies the constant time, and ϑ_i represents the delta parameter.

3.3. Parameter Tuning using Ch-AHO Technique

The proposed ExESNN technique causes high complexity while training with larger data. It may lead to the loss of essential features and be prone to increased error. In this approach, parameters like batch size, learning rate, epochs, and dropout rates are tuned before providing the data into the proposed network model. To perform this, metaheuristic-aided artificial

hummingbird optimizer are utilized to update the model parameters with a globally optimal solution. Hummingbirds (HBs) select feeding sources based on nectar quantity, quality, and replenishing mechanisms. Hummingbirds have exceptional maneuvering abilities and accuracy, which are inspired by hunting for food sources, and are unique in their pursuit of range variation compared to previous techniques. The method's diverse soaring trajectories enhance its mining and exploring capabilities. The visit table simulates a HB's reminiscence for locating sustenance sources. HBs use three hunting tactics and three hovering talents to gather sustenance from various resources. The trio various hovering scenarios such as omnidirectional, oblique, and axial are encompassed to perform the migration, territorial foraging, and guided foraging respectively.

Step 1: Chebyshev Chaos Initialization Phase

Chebyshev Chaos is a phenomenon that, when its initial phase is slightly altered, can show non-linear behavior variations. It can be thought of as a source of randomness and is defined analytically as the unpredictable feature of a basic flexible stochastic system. It is non-repetitive and ergodic, which means that it will search more quickly than arbitrary searches, which heavily rely on coincidence. Hence, the swarm of m HBs are chaotically deliberated to m food resources and it can be mathematically formulated as,

$$u_x = x_{lb} + chebyshev\_choas_k\left(x_{ub} - x_{lb}\right) \qquad w = 1,....,m \tag{17}$$

Here, x_{lb} and x_{ub} indicates the lower and upper boundaries of d^{th} dimension problem, $chebyshev\_choas_k = \cos\left(k\cos^{-1}\left(y_k\right)\right)$ and the position of x^{th} food sources determines the solution of the specific fitness using u_x. The food source with the visit table (VT) can be mathematically formulated as,

$$\beta_{x,e} = \begin{cases} null & if \;\; x = e \quad e = 1,.....,m \\ 0 & if \;\; x \neq e \quad e = 1,.....,m \end{cases} \tag{18}$$

If $x = e$, the VT value $\beta_{x,e}$ attains null shows that the HB is choosing its food from the specific source. In addition to this, when $x \neq e$ the $\beta_{x,e}$ value attains null which signifies the e^{th} food resources where x^{th} HB recently searched at the present iteration.

Step 2: Random Generation

After initiating, arbitrarily choose the most appropriate resolution from the set of input parameters.

Step 3: Fitness Function (FF)

The AH optimizer utilized FF for analyzing the optimality of the proposed classifier model and it is formulated as,

$$f = \min\left(MAE\right), \max\left(R^2\right) \tag{19}$$

Step 4: Guided Searching

Each HB has its common ability to search for food sources within the nectar volume, which determines the accurate source must maximumly replenish the nectar rate and lengthy time period in the absence of any visit. The three various flying scenarios such as omnidirectional, oblique, and axile are encompassed using deliberating the route switch variable during food searching. This vector is defined to blemish one or several d^{th} dimension within the search space. Multiple birds can fly omnidirectionally, but HBs can perform axial glide and diagonal flight. The axile glide can be mathematically formulated as,

$$Z^{(x)} = \begin{cases} 1, & if \;\; x = randk\left(\left[1, d\right]\right) \\ null, & otherwise \end{cases} \qquad x = 1, ..., d \tag{20}$$

The oblique flight can be mathematically formulated in the equation (21),

$$Z^{(x)} = \begin{cases} 1, & if \;\; x = pop\left(i\right), i \in \left[1, l\right] \\ & pop = randperm\left(\delta\right) \\ & \delta \in \left[2, \left[r_1 \times \left(d-2\right)\right] + 1\right] \\ 0, & otherwise \;\; x = 1, ..., d \end{cases} \tag{21}$$

The formulation for omnidirectional travel can be expressed as given below:

$$Z^{(x)} = 1 \quad j = 1,....,d \tag{22}$$

Here, $Z^{(x)}$ indicates the random integer ranges between 1 and d returns $randk$. The random permutation integer sequence between 1 and knowledge percentage (KP) is obtained using $randperm(\delta)$.Moreover, r_1 indicates the arbitrary value ranges between 0 and 1. Hence, the food resources are enhanced based on final food resources which are recognized from the present resources. Finally, iterative directed searching can be mathematically formulated in equation (23),

$$VP_x = z_{x,Tar}(t_p) + b \times Z \times \left(z_x(t_p) - z_{x,Tar}(t_p)\right) \tag{23}$$

Here, $z_x(t_p)$ signifies the site of x^{th} sustenance resource at the present iteration, $z_{x,Tar}(t_p)$ symbolizes the position of food source that x^{th} HB determined to eat and indicated the normal distribution with zero mean value and SD of 1. The above equation (23) makes every current resource change its position concerning the target food resources and iterates the guided searching in HBs by different flying patterns. The position of x^{th} food resources can be mathematically formulated in equation (24),

$$z_x(t_p+1) = \begin{cases} z_x(t_p) & if \ \ f\left(z_x(t_p)\right) \le \left(VP_x(t_p+1)\right) \\ u_x(t_p+1) & if \ f\left(z_x(t_p)\right) > \left(VP_x(t_p+1)\right) \end{cases} \tag{24}$$

Here, f indicates the FF. The VT is the essential element of the AHO technique that recovers the data about the food source visits. The VT stores the undiscovered time interval of every food resource which deliberates the maximum visit degree.

Step 5: Territorial Searching

If the flower nectar has been fatigued, the HBs elucidate the new food resources when compared to the visit of other present food resources. Hence, the HBs easily travel to a nearby location in their own area, where the updated food resources are determined. The territorial searching of HBs can be mathematically formulated in equation (25),

$$z_x(t_p+1) = z_x(t_p) + B_p \times Z \times z_x(t_p) \quad B_p \sim N(0,1) \tag{25}$$

The parameter B_p indicates the territorial searching that has the mean zero and SD of 1 and defines the normal distribution.

Step 6: Migration Searching

When the present iteration diminishes the existing mentioned migration coefficient parameter, the bird at the sustenance resource with the minimum replenishment nectar rate will randomly define the new food resources within the territory. The migratory behavior of HBs can be mathematically formulated in equation (26),

$$z_{wor}\left(t_p+1\right)=lb+rand\left(ub-lb\right) \tag{26}$$

Here, z_{wor} signifies the food resources with a minimal replenishment nectar rate. The migration coefficient t_p according to the m population size can be formulated in equation (27),

$$t_p=2m \tag{27}$$

Based on equation (21), the starting stages of repetition, the substantial distance between resources of foods is accomplished in the exploration phase. However, when the repetition increases, distance minimizes and hence, more priority is given to the exploitation phase.

Step 7: Return the Best Optimal Solution

Step 8: Termination

Algorithm 1: Pseudocode of Ch-AHO technique
Define $M_{pop}=m=Population\,size$ **Define** $M_{iter,\max}$ **Define** maximum and minimum population bounds **Initialize** population based on chebyshev chaos using equation (17), **While** $t_p \le M_{iter,\max}$ **for** every population evaluate switch direction vector Z if $rand \le 1/3$ Track oblique flight using equation (21), **else if** $rand \le 2/3$ Track oblique flight using equation (22), **else** Track axial fight using equation (20),

end if
end for
for every population, the foraging behaviour is updated
if $rand \leq 0.5$
Track guided searching using equations (20) to (26),
else if Track territorial searching using equations (27),
end if
if $t_p = 2m$
Track migration searching using equation (28),
end if
end for
Improve locations
Return the best optimal solution
$t_p = t_p + 1$
end while

4. Results and Discussion

The developed framework is processed and experimented with via the Python platform. The Chennai Floods dataset on Kaggle contains data regarding the deluge that hit the Indian state of Chennai in December 2020. The dataset contains details of the relief camps involved, areas affected, rainfall data, and resources required related to this flood crisis. This dataset will be very instrumental in the analysis and research for understanding the effects of the floods and areas for improvement on disaster response and management, besides the predictive models of future preparedness. Both Kling-Gupta Efficiency (KGE) and Nash-Sutcliffe Efficiency (NSE) are statistical metrics used to evaluate the performance of models, particularly in hydrology, environmental sciences, and machine learning applications related to time-series predictions. Suresh Subramanian et al. (2024) 80% and 20% of the data are considered for the training and testing process respectively in the ratio of 8:2.

4.1. Computational Metrics

Assessment measurements like R2, RMSE, KGE, NSE, and MAE are inspected to comprehend the developed method effectiveness.

4.1.1. KGE Analysis

It is demarcated as the degree to which the effectiveness that has been yielded by the developed model is determined between forecasted and modeled hydrological data points. This can be expressed by Venkatraman, M et al (2024),

$$KGE = 1 - \sqrt{(i-1)^2 (\alpha - 1)^2 (\mu - 1)^2} \tag{29}$$

Here, α signifies the standard deviation of predicted and actual values, μ signifies the ratio of the mean of predicted and actual values, and i indicates the correlation coefficient between forecasted and modeled data.

4.1.2. MAE Analysis

It is demarcated as the measure of the average absolute difference between actual and forecasted values and this can be formulated as depicted below:

$$MAE = \frac{\sum_{i=1}^{u} |v_i - v_i'|}{u} \tag{30}$$

Here, u indicates the total data points.

4.1.3. NSE Analysis*

It is the degree used in hydrological modeling to evaluate the efficacy of the developed model similar to the observed value. It can be formulated as,

$$NSE = 1 - \frac{\sum_{i=1}^{u} (v_i' - v_i)^2}{\sum_{x=1}^{z} (v_i - \tilde{v}_i)^2} \tag{31}$$

4.1.4. R2 Analysis

It measures the amount of variance of a dependent variable accounted for by an independent variable in a regression model. This is mathematically expressed as:

$$R^2 = \frac{\sum_{x=1}^{u} (v_i - v')(x_i - x')}{\sqrt{\sum_{i=1}^{u} ((v_i - v'))^2} \sqrt{\sum_{i=1}^{u} (x_i - x')^2}} \tag{32}$$

4.1.5. RMSE Analysis

It is the average of the sum of squares of the differences between observed and forecasted values. Mathematically, it can be expressed as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{u}\left(v_i - v_i'\right)^2}{u}} \tag{33}$$

Here, $\tilde{v}_i$ represents the mean of all the values, v_i' indicates the predicted value, and v_i indicates the actual values.

4.2. Computational Analysis of Developed Scheme over Conventional Studies

In this section, the performance evaluation of the proposed method is presented through graphical representation. This has been evaluated against several traditional methodologies, namely ENN, GWO-SNN, AHO-ESNN, PSO-LSTM, and some other previously proposed existing methodologies. Results are elaborated and discussed as follows.

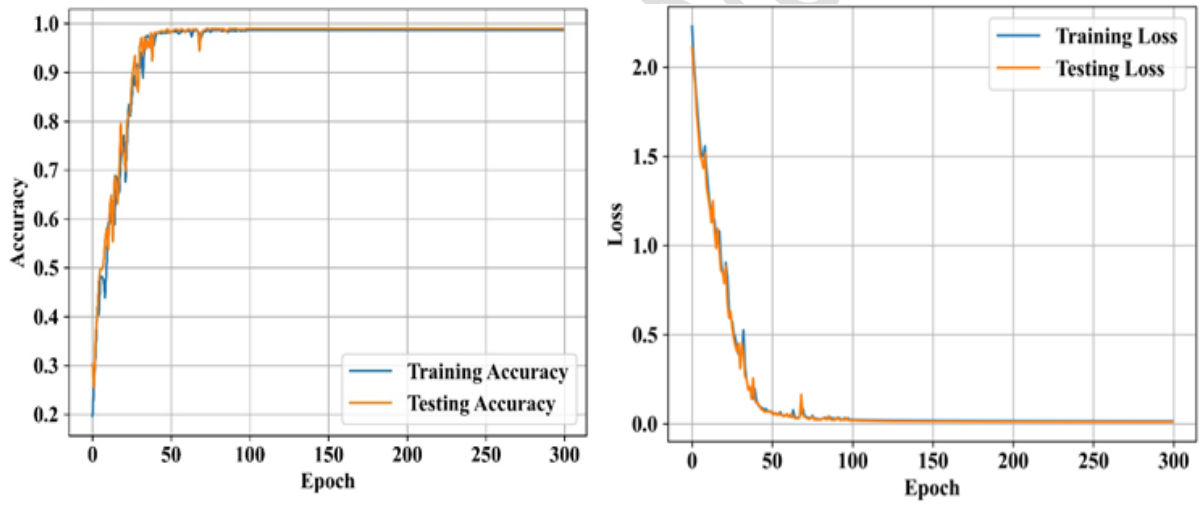

Figure 3. Accuracy and Loss analysis by altering epochs

Figure 3 signifies Accuracy and Loss as a function of epochs. The graphical illustration shows performance on train data and test data for 300 epochs. For 300 epochs, training accuracy was 99.3 percent and testing accuracy was 98.74 percent. The training loss was about 0.02 and testing loss was around 0.045 percent.

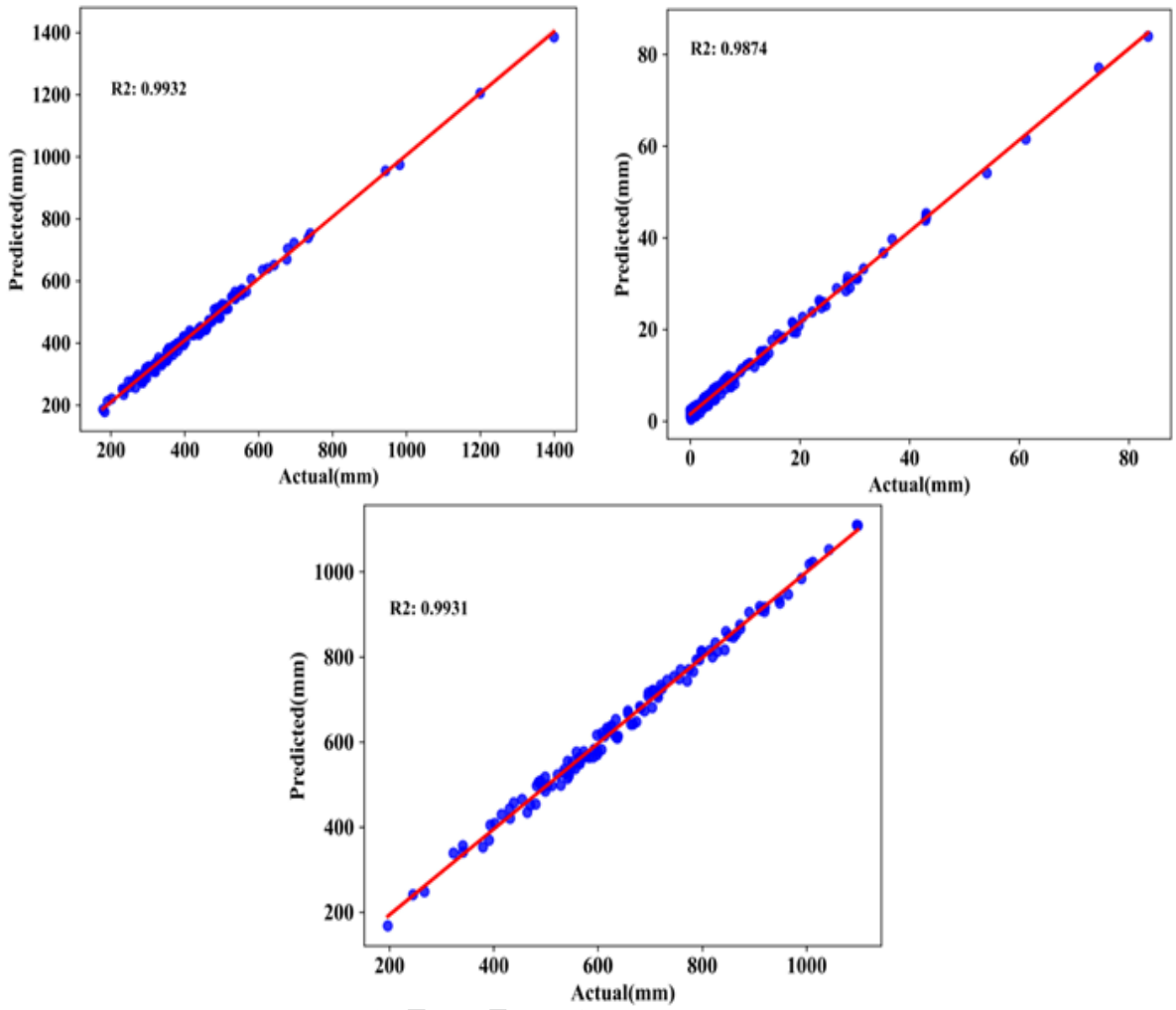

Figure 4. Actual and Predicted Rainfall for varying months, (a) November, (b) December and (c) January

Figures 4(a-c) represent the real and the predicted flooded conditions for three months ((a) November, (b) December and (c) January) for all years over the Chennai state. The performance of the Ch-AHO-ExESNN technique is discussed in the analysis below. From these graphs, it can be noticed that the model very finely fits with the actual data, hence proving to be efficient in flood prediction. The developed method becomes efficient and works well when trained with huge data pertaining to floods.

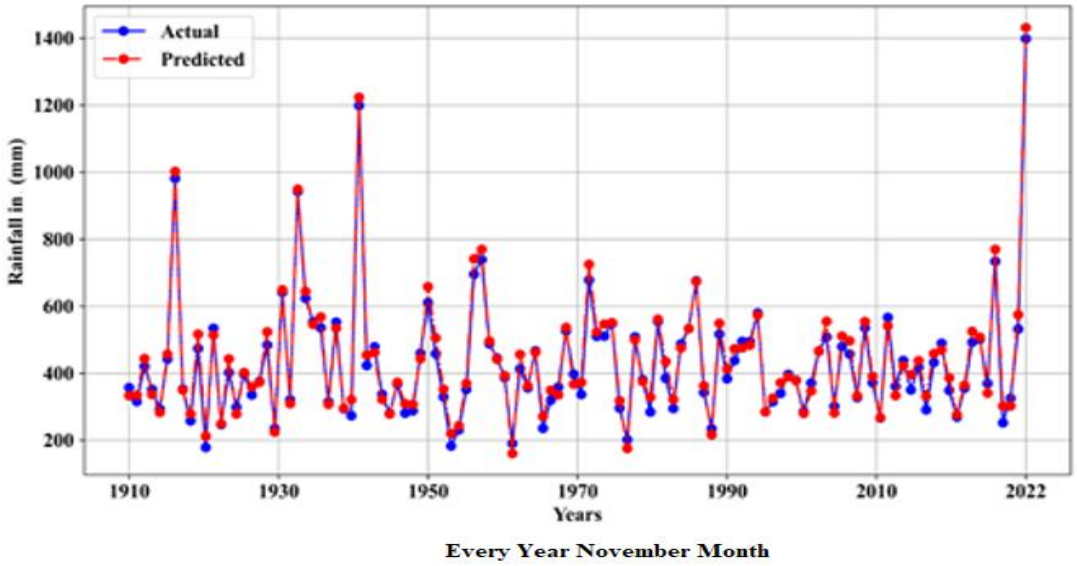

Figure 5. Actual and Predicted flood for varying years, (a) November

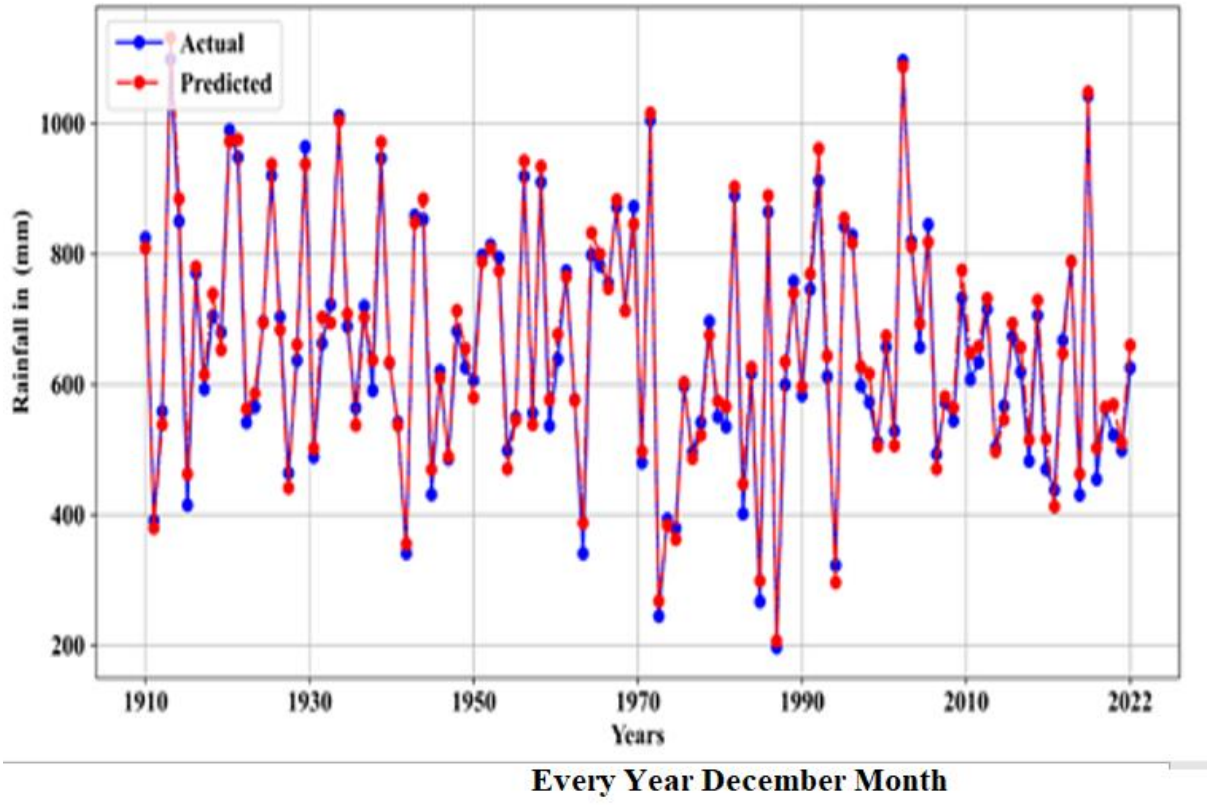

Figure 5. Actual and Predicted flood for varying years (b) December

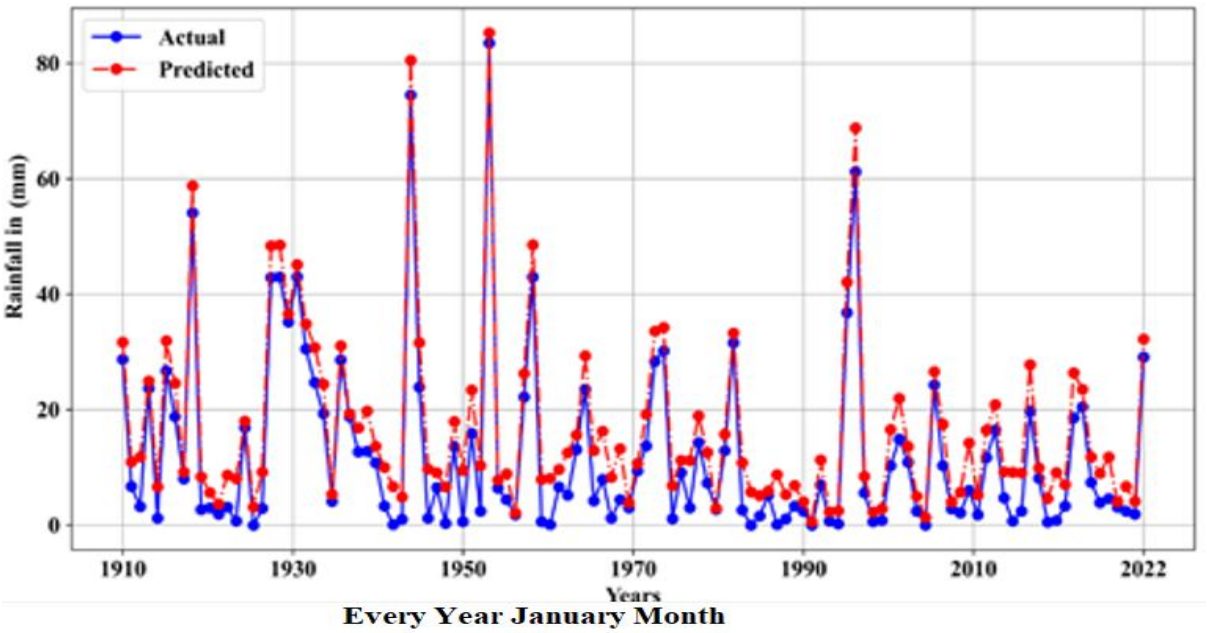

Figure 5. Actual and Predicted flood for varying years (c) January

Figures 5(a-c) illustrate the actual versus predicted flood conditions for different years in (a) November, (b) December and (c) January. This analysis assesses the performance of the Ch-AHO-ExESNN technique. The graphs demonstrate that the model closely aligns with the actual data, indicating its effectiveness in flood prediction. The developed method shows strong performance, even when trained on extensive flood datasets.

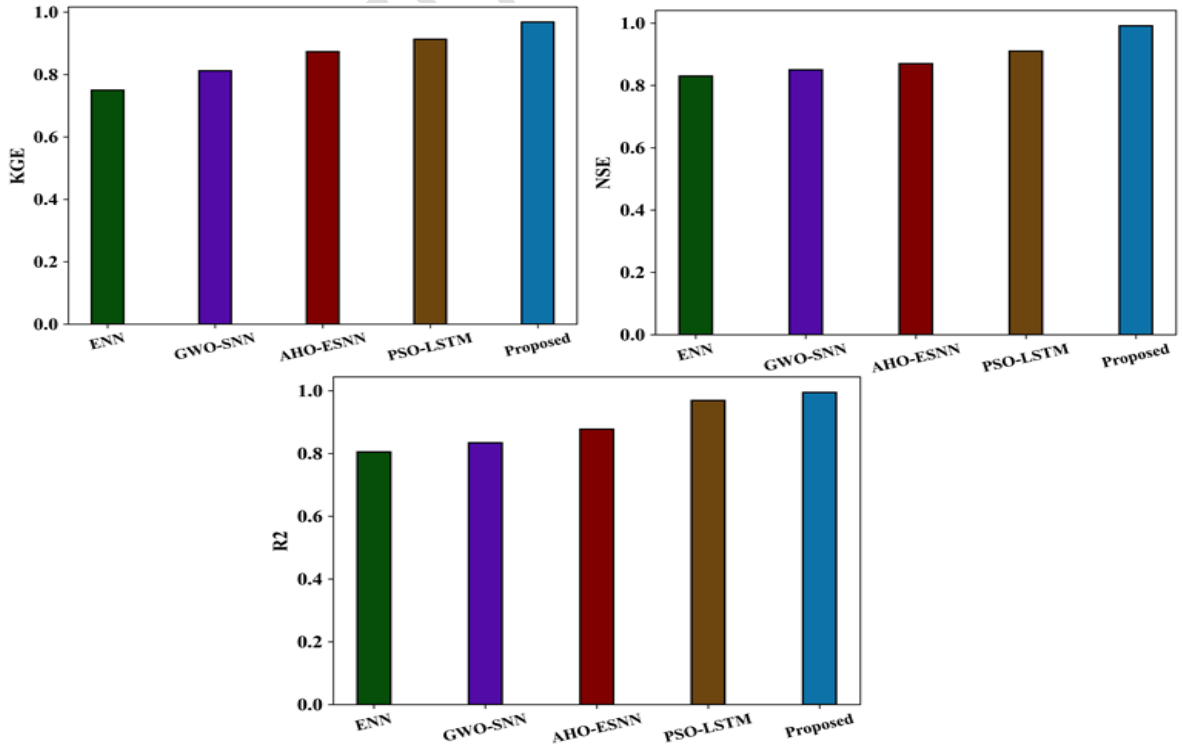

Figure 6. KGE, NSE, and R2 analysis under different traditional frameworks

Figures 6(a-c) present the analysis of KGE, NSE, and R^2 metrics across various traditional frameworks. The graphical results indicate that the Ch-AHO-ExESNN technique achieves minimal error and superior performance compared to conventional methods, demonstrating its suitability for future weather prediction tasks. Specifically, for KGE, the values obtained were 0.75 for ENN, 0.812 for GWO-SNN, 0.873 for AHO-ESNN, 0.913 for PSO-LSTM, and 0.968 for the Ch-AHO-ExESNN technique. For NSE, the values were 0.83 for ENN, 0.85 for GWO-SNN, 0.87 for AHO-ESNN, 0.91 for PSO-LSTM, and 0.991 for the Ch-AHO-ExESNN technique. For R^2, the values were 0.805 for ENN, 0.834 for GWO-SNN, 0.877 for AHO-ESNN, 0.969 for PSO-LSTM, and 0.994 for the Ch-AHO-ExESNN technique. These results highlight that the Ch-AHO-ExESNN model is well-suited to support future real-time weather prediction applications.

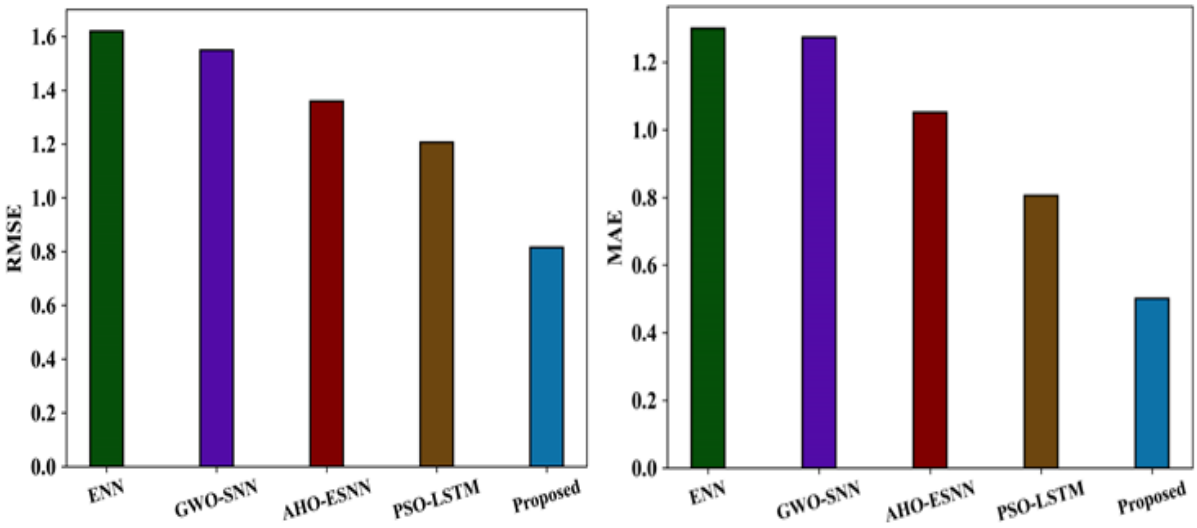

Figure 7. MAE and RMSE analysis under different traditional frameworks

Figure 7 displays the analysis of MAE and RMSE across different traditional frameworks. The results demonstrate that the CResAtt-GCU technique achieves minimal error compared to conventional methods, indicating its potential for future weather prediction applications. Specifically, for MAE, the values were 1.3 for ENN, 1.2737 for GWO-SNN, 1.0516 for AHO-ESNN, 0.8059 for PSO-LSTM, and 0.50132 for the CResAtt-GCU technique. For RMSE, the values were 1.62 for ENN, 1.55 for GWO-SNN, 1.36 for AHO-ESNN, 1.207 for PSO-LSTM, and 0.815 for the CResAtt-GCU technique. These results highlight the superior performance of the CResAtt-GCU model in minimizing error for weather prediction processes.

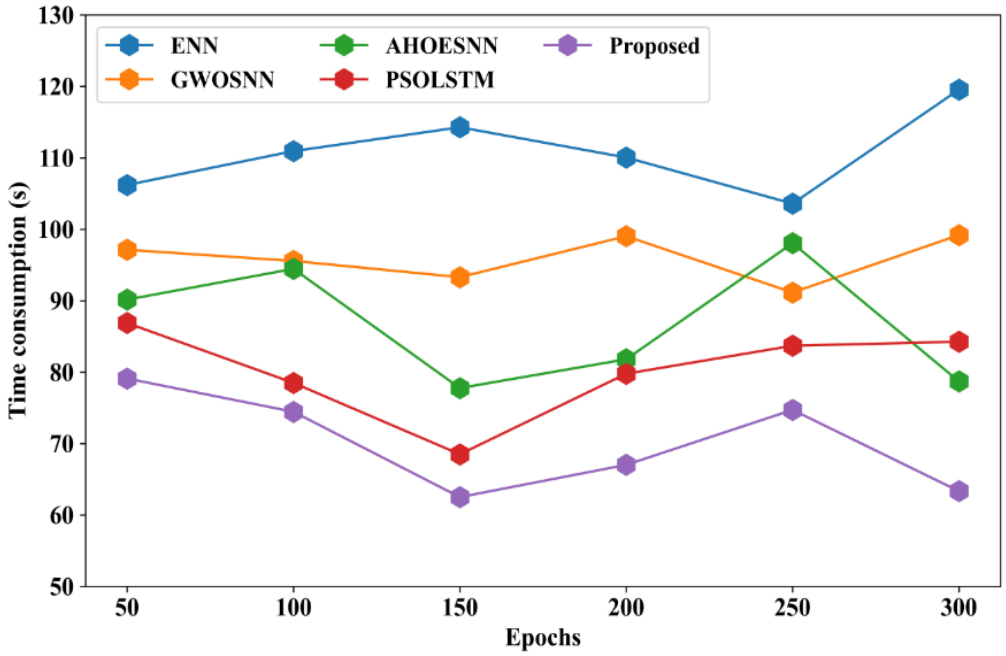

Figure 8. Computational Time analysis by altering epochs

Figure 8 illustrates the time complexity analysis for varying epochs across different techniques. The graph shows that the developed method achieves lower time complexity compared to traditional approaches. Conventional techniques tend to consume more time when training with complex flood data, which can increase model error due to challenges with data distribution. In contrast, the developed method addresses these issues effectively, delivering accurate results with reduced training time. For 300 epochs, the time taken by existing techniques was 119.54s for ENN, 99.18s for GWO-SNN, 78.70s for AHO-ESNN, 84.25s for PSO-LSTM, and 63.33s for the Ch-AHO-ExESNN technique. For 50 epochs, the times recorded were 106.20s for ENN, 97.12s for GWO-SNN, 90.15s for AHO-ESNN, 86.87s for PSO-LSTM, and 79.10s for the CResAtt-GCU technique.

5. Conclusion

In this framework, a new chaotic artificial hummingbird optimizer (Ch-AHO) was proposed and examined along with the extended Elman Spiking neural network (ExESNN) technique in the flood prediction problem in various regions of Chennai. The proposed optimizer proved to be very effective in tuning the model parameters with less computational time and more enhanced global optimal solutions. Besides, the ExESNN technique gives an effective flood prediction with a minimal computational complication by using the adaptive optimal feature learning mechanism. Another technique that is introduced is MAxAbsScaler (MAS) for normalizing the flood data by removing unwanted missing values to make the effectiveness

and classification process of flood prediction more effective. In this work, the proposed framework is tested on the Python platform and real-time flood data dissemination datasets are collected and processed. Various computational measures like R2, RMSE, KGE, NSE, MAE, and CT have been considered, and the proposed efficacy has been proved with different conventional frameworks. It returns an overall R2 of 0.994, RMSE of 0.851, KGE of 0.968, and NSE of 0.991 with a MAE of 0.50 and an overall CT of 63.33s for the developed technique against various conventional studies on flood forecasting in different regions of Chennai. However, the developed framework relies significantly on historical climatological data, which may not always accurately reflect current weather patterns due to the impacts of climate change. The accuracy of flood forecasts can also be influenced by unforeseen weather events or alterations in local environmental conditions. Future research will extend this work by integrating additional environmental factors, such as humidity and wind patterns, to better understand their effects on flood dynamics in various regions. Moreover, upcoming studies will include exploring different urban areas or countries to gain a more comprehensive understanding of how temperature variations and other factors interact to influence flood conditions.

Data Availability Statement

1. https://data.opencity.in/dataset/chennai-floods-2020-data
2. https://data.opencity.in/dataset/resettled-in-the-paths-of-floods December 2023
3. https://www.kaggle.com/datasets/monkeyddonut/chennai-urban-flood-map-2015-raster-tif

References

Adnan, Mohammed Sarfaraz Gani, Zakaria Shams Siam, Irfat Kabir, Zobaidul Kabir, M. Razu Ahmed, Quazi K. Hassan, Rashedur M. Rahman, and Ashraf Dewan. "A novel framework for addressing uncertainties in machine learning-based geospatial approaches for flood prediction." Journal of Environmental Management 326 (2023): 116813.

Alotaibi Y., Deepa R., Shankar K. and Rajendran S. (2024), Inverse chi-square-based flamingo search optimization with machine learning-based security solution for Internet of Things edge devices. AIMS Math, 9, 22-37.

Alotaibi Y., Rajendran B., and Rajendran S. (2024), Dipper throated optimization with deep convolutional neural network-based crop classification for remote sensing image analysis. PeerJ Computer Science, 10, 1828.

Arun Mozhi Selvi Sundarapandi, Sundara Rajulu Navaneethakrishnan, Hemlathadhevi A, Surendran Rajendran, (2024) “A Light weighted Dense and Tree structured simple recurrent unit (LDTSRU) for flood prediction using meteorological variables”, Global NEST Journal. Available at: https://doi.org/10.30955/gnj.06242.

Aydin, Halit Enes, and Muzaffer Can Iban. "Predicting and analyzing flood susceptibility using boosting-based ensemble machine learning algorithms with SHapley Additive exPlanations." Natural Hazards 116, no. 3 (2023): 2957-2991.

Babu T, Raveena Selvanarayanan, Tamilvizhi Thanarajan and Surendran Rajendran* (2024) “Integrated Early Flood Prediction using Sentinel-2 Imagery with VANET-MARL-based Deep Neural RNN”, Global NEST Journal, 26(10). https://doi.org/10.30955/gnj.06554

Dehghani, Adnan, Hamza Mohammad Zakir Hiyat Moazam, Fatemehsadat Mortazavizadeh, Vahid Ranjbar, Majid Mirzaei, Saber Mortezavi, Jing Lin Ng, and Amin Dehghani. "Comparative evaluation of LSTM, CNN, and ConvLSTM for hourly short-term streamflow forecasting using deep learning approaches." Ecological Informatics 75 (2023): 102119.

Dtissibe, Francis Yongwa, Ado Adamou Abba Ari, Hamadjam Abboubakar, Arouna Ndam Njoya, Alidou Mohamadou, and Ousmane Thiare. "A comparative study of Machine Learning and Deep Learning methods for flood forecasting in the Far-North region, Cameroon." Scientific African 23 (2024): e02053.

Farahmand, Hamed, Yuanchang Xu, and Ali Mostafavi. "A spatial–temporal graph deep learning model for urban flood nowcasting leveraging heterogeneous community features." Scientific Reports 13, no. 1 (2023): 6768.

He, Jian, Limin Zhang, Te Xiao, Haojie Wang, and Hongyu Luo. "Deep learning enables super-resolution hydrodynamic flooding process modeling under spatiotemporally varying rainstorms." Water Research 239 (2023): 120057.

Li, Wenzhong, Chengshuai Liu, Yingying Xu, Chaojie Niu, Runxi Li, Ming Li, Caihong Hu, and Lu Tian. "An interpretable hybrid deep learning model for flood forecasting based on Transformer and LSTM." Journal of Hydrology: Regional Studies 54 (2024): 101873.

Lin, Chu-Hsuan Abraham, Chen-Yu Liu, and Kuan-Cheng Chen. "Quantum-Train Long Short-Term Memory: Application on Flood Prediction Problem." arXiv preprint arXiv:2407.08617 (2024).

Lin, Lin, Chaoqing Tang, Qiuhua Liang, Zening Wu, Xinling Wang, and Shan Zhao. "Rapid urban flood risk mapping for data-scarce environments using social sensing and region-stable deep neural network." Journal of Hydrology 617 (2023): 128758.

Moon, Hyeontae, Sunkwon Yoon, and Youngil Moon. "Urban flood forecasting using a hybrid modeling approach based on a deep learning technique." Journal of Hydroinformatics 25, no. 2 (2023): 593-610.

Nguyen, Huu Duy, Chien Pham Van, and Anh Duc Do. "Application of hybrid model-based deep learning and swarm-based optimizers for flood susceptibility prediction in Binh Dinh province, Vietnam." Earth Science Informatics 16, no. 2 (2023): 1173-1193.

Ouma, Yashon O., and Lawrence Omai. "Flood Susceptibility Mapping Using Image-Based 2D-CNN Deep Learning: Overview and Case Study Application Using Multiparametric

Spatial Data in Data-Scarce Urban Environments." International Journal of Intelligent Systems 2023, no. 1 (2023): 5672401.

Rahman, Tanvir, Miah Mohammad Asif Syeed, Maisha Farzana, Ishadie Namir, Ipshita Ishrar, Meherin Hossain Nushra, and Bhoktear Mahbub Khan. "Flood prediction using ensemble machine learning model." In 2023 5th International congress on human-computer interaction, Optimization and Robotic Applications (HORA), pp. 1-6. IEEE, 2023.

Shikhteymour, Sharareh Rashidi, Moslem Borji, Mehdi Bagheri-Gavkosh, Ehsan Azimi, and Timothy W. Collins. "A novel approach for assessing flood risk with machine learning and multi-criteria decision-making methods." Applied geography 158 (2023): 103035.

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. Comput. Syst. Sci. Eng., 46, 835-851.

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, 2021, Springer International Publishing, 4, 94-111.

Suresh Subramanian, Geetha Rani K, Maheswari Madhavan, Surendran Rajendran . (2024) “An Automatic Data-Driven Long-term Rainfall Prediction using Humboldt Squid Optimized Convolutional Residual Attentive Gated Circulation Model in India”. Global NEST Journal, Vol 26, No 10, 06421.

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, 3-20.

Venkatraman, M., Surendran R., Senduru Srinivasulu, Vijayakumar K. (2024) “Water quality prediction and classification using Attention based Deep Differential RecurFlowNet with Logistic Giant Armadillo Optimization”, Global NEST Journal. https://doi.org/10.30955/gnj.0679.

Wang, Huiliang, Shanlun Xu, Hongshi Xu, Zening Wu, Tianye Wang, and Chao Ma. "Rapid prediction of urban flood based on disaster-breeding environment clustering and Bayesian optimized deep learning model in the coastal city." Sustainable Cities and Society 99 (2023): 104898.

Weng, Peiyao, Yu Tian, Yingfei Liu, and Ying Zheng. "Time-series generative adversarial networks for flood forecasting." Journal of Hydrology 622 (2023): 129702.

Windheuser, L., R. Karanjit, R. Pally, S. Samadi, and N. C. Hubig. "An end-to-end flood stage prediction system using deep neural networks." Earth and Space Science 10, no. 1 (2023): e2022EA002385.

Wu, Junhao, Zhaocai Wang, Jinghan Dong, Xuefei Cui, Sen Tao, and Xi Chen. "Robust runoff prediction with explainable artificial intelligence and meteorological variables from deep learning ensemble model." Water Resources Research 59, no. 9 (2023): e2023WR035676.

Xu, Chengjing, Pingan Zhong, Feilin Zhu, Luhua Yang, Sen Wang, and Yiwen Wang. "Real-time error correction for flood forecasting based on machine learning ensemble method and its uncertainty assessment." Stochastic Environmental Research and Risk Assessment 37, no. 4 (2023): 1557-1577.

Zhong, Pengcheng, Yueyi Liu, Hang Zheng, and Jianshi Zhao. "Detection of urban flood inundation from traffic images using deep learning methods." Water Resources Management 38, no. 1 (2024): 287-301.

Zhong, Ming, Hongrui Zhang, Tao Jiang, Jun Guo, Jinxin Zhu, Dagang Wang, and Xiaohong Chen. "A hybrid model combining the Cama-Flood Model and Deep Learning methods for Streamflow Prediction." Water Resources Management 37, no. 12 (2023): 4841-4859.

Zhou, Qianqian, Shuai Teng, Zuxiang Situ, Xiaoting Liao, Junman Feng, Gongfa Chen, Jianliang Zhang, and Zonglei Lu. "A deep-learning-technique-based data-driven model for accurate and rapid flood predictions in temporal and spatial dimensions." Hydrology and Earth System Sciences 27, no. 9 (2023): 1791-1808.

Zhang, Lin, Huapeng Qin, Junqi Mao, Xiaoyan Cao, and Guangtao Fu. "High temporal resolution urban flood prediction using attention-based LSTM models." Journal of Hydrology 620 (2023): 129499.