atmosphere

MDPI

Article

Enhancing Cyclone Intensity Prediction for Smart Cities Using a Deep-Learning Approach for Accurate Prediction

Senthil Kumar Jayaraman [1], Venkataraman Venkatachalam [2], Marwa M. Eid [3,*], Kannan Krithivasan [2], Sekar Kidambi Raju [4,*], Doaa Sami Khafaga [5], Faten Khalid Karim [5] and Ayman Em Ahmed [6]

1 Department of CSE, SRC, SASTRA Deemed University, Kumbakonam 612001, India; senthilkumarcse@src.sastra.edu
2 Department of Mathematics, SASTRA Deemed University, Kumbakonam 613401, India; mathvvr@maths.sastra.edu (V.V.); deankannan@sastra.edu (K.K.)
3 Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 35712, Egypt
4 School of Computing, SASTRA Deemed University, Thanjavur 613401, India
5 Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; dskhafga@pnu.edu.sa (D.S.K.); fkdiaaldin@pnu.edu.sa (F.K.K.)
6 Faculty of Engineering, King Salman International University, El-Tor 8701301, Egypt; ayman.ahmed@alumni.carleton.ca
* Correspondence: mmm@ieee.org (M.M.E.); sekar_kr@cse.sastra.ac.in (S.K.R.)

Abstract: Accurate cyclone intensity prediction is crucial for smart cities to effectively prepare and mitigate the potential devastation caused by these extreme weather events. Traditional meteorological models often face challenges in accurately forecasting cyclone intensity due to cyclonic systems' complex and dynamic nature. Predicting the intensity of cyclones is a challenging task in meteorological research, as it requires expertise in extracting spatio-temporal features. To address this challenge, a new technique, called linear support vector regressive gradient descent Jaccardized deep multilayer perceptive classifier (LEGEMP), has been proposed to improve the accuracy of cyclone intensity prediction. This technique utilizes a dataset that contains various attributes. It employs the Herfindahl correlative linear support vector regression feature selection to identify the most important characteristics for enhancing cyclone intensity forecasting accuracy. The selected features are then used in conjunction with the Nesterov gradient descent jeopardized deep multilayer perceptive classifier to predict the intensity classes of cyclones, including depression, deep depression, cyclone, severe cyclone, very severe cyclone, and extremely severe cyclone. Experimental results have demonstrated that LEGEMP outperforms conventional methods in terms of cyclone intensity prediction accuracy, requiring minimum time, error rate, and memory consumption. By leveraging advanced techniques and feature selection, LEGEMP provides more reliable and precise predictions for cyclone intensity, enabling better preparedness and response strategies to mitigate the impact of these destructive storms. The LEGEMP technique offers an improved approach to cyclone intensity prediction, leveraging advanced classifiers and feature selection methods to enhance accuracy and reduce error rates. We demonstrate the effectiveness of our approach through rigorous evaluation and comparison with conventional prediction methods, showcasing significant improvements in prediction accuracy. Integrating our enhanced prediction model into smart city disaster management systems can substantially enhance preparedness and response strategies, ultimately contributing to the safety and resilience of communities in cyclone-prone regions.

Keywords: cyclone intensity prediction; Herfindahl correlative linear support vector regression; disaster management; smart city

Citation: Jayaraman, S.K.; Venkatachalam, V.; Eid, M.M.; Krithivasan, K.; Raju, S.K.; Khafaga, D.S.; Karim, F.K.; Ahmed, A.E. Enhancing Cyclone Intensity Prediction for Smart Cities Using a Deep-Learning Approach for Accurate Prediction. *Atmosphere* **2023**, *14*, 1567. https://doi.org/10.3390/atmos14101567

Academic Editor: Yubin Li

Received: 11 August 2023
Revised: 7 October 2023
Accepted: 12 October 2023
Published: 16 October 2023

1. Introduction

In recent decades, the increasing frequency and intensity of cyclones have posed significant challenges to communities and infrastructure in coastal regions worldwide.

These extreme weather events jeopardize human lives and cause substantial damage to the built environment, leading to economic losses and disruptions in daily life. To mitigate the adverse effects of cyclones and improve disaster preparedness, accurate cyclone intensity prediction is paramount. The main aim of accurate estimation of cyclone intensity is to forecast the performance of intensity. Initializing prediction models and tragedy management in the meteorological industry is essential. However, accurate and timely intensity assessment is a challenge because it requires domain knowledge to automatically remove the structure features and parameters.

A convolutional long short-term memory (ConvLSTM) network was performed by Tong et al. [1] to extract both time-related correlations of different feature parameters for the short-term prediction of a tropical cyclone. However, it needed to develop recursive models for cyclone track and intensity. The authors used month-based weather data gathered from four Malaysian weather stations during the 2000–2019 timeframe to train and evaluate the models. Several input attributes (predictor variables) were investigated to select the most suitable variables for the machine-learning models [2]. The climate indices were utilized as input attributes to train and evaluate the models. The artificial neural network (ANN) models were compared with the traditional statistical models, and the results showed that the ANN models outperformed the statistical models in forecasting long-term streamflow rates. The study concluded that the ANN models can be a reliable tool for long-term streamflow forecasting [3]. The wavelet neural network

(WNN) scheme was compared with the traditional statistical models, and the results showed that the WNN scheme outperformed the statistical models in predicting seasonal rainfall with a one-year lead time. The study concluded that the WNN scheme can be a reliable tool for predicting seasonal rainfall with a one-year lead time [4].

An XGBoost and a decision-tree-based machine-learning algorithm were designed by Chan et al. [5] to improve cyclone intensity prediction. However, the prediction of cyclone intensity took more time. In Lee et al. [6], machine-learning techniques were introduced to estimate the intensity of tropical cyclones by leveraging both spatial and temporal characteristics. However, deep-learning models were not used to develop the estimation of cyclone intensity prediction. A three-dimensional convolutional neural network (3D-CNN) was determined by Wang et al. [7] for tropical cyclone (TC) intensity changes using the dual spatial features. However, they removed additional fused features to allow for prediction over an extended time. Deep-learning (DL)-based multilayer perceptron (MLP) was developed by Wenwei Xu et al. [8] for tropical intensity prediction. However, higher true detections were not obtained.

Giffard-Roisin et al. [9] developed a neural network model for tropical cyclone forecasting. However, the performance of the error rate of tropical cyclone forecasting was not minimized. In Jiang et al. [10], a typhoon intensity spatiotemporal prediction network (TITP-Net) model was introduced to fully extract the information from the datasets by measuring the spatiotemporal dependencies. However, the short-term prediction of typhoon intensity was not focused.

Research contributions: The work presented in this article aimed to improve cyclone prediction accuracy to address the disadvantages mentioned above by establishing the following novel contributions.

- A new linear support vector regressive gradient descent Jaccardized deep multilayer perceptive classifier (LEGEMP) is developed to increase the accuracy of cyclone intensity level forecasts using various strategies, such as feature selection and classification.
- LEGEMP partitions the input information into relevant and irrelevant feature sets using Herfindahl correlative linear support vector regression to minimize cyclone intensity prediction time.
- The Nesterov gradient descent Jaccardized deep multilayer perceptive classifier is used in LEGEMP to analyze feature selection using the Jaccard similarity index. The soft step activation function provides the output results. Finally, the Nesterov gradient

descent approach updates the weights and reduces the cyclone intensity forecast error rate.

Structure of this paper: This paper is outlined as follows. Section 2 reviews the related works. Section 3 describes the proposed LEGEMP with a different process for cyclone intensity prediction. In Section 4, the experimental assessment and dataset description are presented. Section 5 provides the simulation results and discussion with certain parameters. Finally, Section 6 is the conclusion of the paper.

2. Related Works

Lian et al. [11] proposed a data-driven, deep-learning approach for forecasting tropical cyclone trajectories based on spatial and atmospheric factors. However, it failed to enhance the model by considering several tropical cyclones. Higa et al. [12] developed a deep-learning approach to increase cyclone intensity forecast accuracy by considering meteorological domain knowledge. It did not, however, create an effective deep-learning architecture for predicting typhoon severity using additional observation data.

Zhou et al. [13] proposed a logistic-growth-equation technique for predicting the strength of tropical cyclones. However, it did not increase the accuracy of tropical cyclone intensity forecasting. Sridevi et al. [14] created the Global Forecast System (GFS) to forecast tropical storm course and intensity for 2019 and 2020. However, the various error statistics findings were not minimized. Sarkar et al. [15] proposed a neuro-computing-based adaptive intelligent system to forecast cyclones over the Bay of Bengal. However, making an accurate prognosis was a difficult challenge. Wahiduzzaman et al. [16] developed a regional-scale spatial statistical approach to identify the relationship between tropical cyclones and sea surface temperature. However, it did not concentrate on tropical cyclone strength prediction using a generalized additive model.

Singh et al. [17] reported an evaluation of the model for forecasting exceptionally severe cyclonic storms in the Bay of Bengal area. Tian et al. [18] used a CNN model for tropical storm strength estimation using satellite remote sensing data. Devaraj et al. [19] created an enhanced deep convolutional neural network (CNN) to determine cyclone strength with decreased root mean squared error. However, the prediction model's efficiency was not increased. ManilMaskey et al. [20] developed a deep-learning tropical storm intensity rating. However, it failed to consider the comprehensive investigation of a specific storm to comprehend model performance with structural storm modifications.

The system uses data from diverse sources, including soil, plant condition, environmental sensor networks, meteorological predictions, and high-resolution UAV and Satellite imagery, to provide farmers with a dynamic and up-to-date visualization of their agricultural domains [21]. These systems use sensory systems and data from diverse sources, including weather forecasts, soil moisture sensors, and plant water stress sensors, to optimize irrigation schedules and reduce water waste. They also provide farmers with real-time information on crop water requirements, soil moisture levels, and other critical parameters, enabling them to make informed decisions about irrigation management [22]. The system uses a combination of sensors, including electromagnetic induction, time-domain reflectometry, and capacitance sensors, to collect data on soil properties like texture, structure, and water content. The data is then fed into a deep learning model that predicts soil moisture levels with high accuracy [23].

3. Methodology

Cyclones are well-known natural catastrophes and intense weather events that inflict significant damage and loss of life. Predicting intensity properly and efficiently is critical in wind engineering to prevent and mitigate cyclone-induced disasters. However, inconsequential cyclone power or wind speed phases become difficult to assess and, as a result, tough to anticipate with greater accuracy and in less time using current approaches. Then, a unique deep-learning algorithm, known as LEGEMP, correctly and swiftly forecasts cyclone intensity.

Figure 1 above describes the architecture diagram of the proposed LEGEMP technique that consists of two major processes, namely attribute selection and classification, for developing the accuracy of cyclone intensity with big data. At first, the dataset comprises many features $f = f_1,\ f_2,\ f_3 \ldots, f_n$ and data $D\ \in B_1,\ B_2,\ \ldots,\ B_n$.

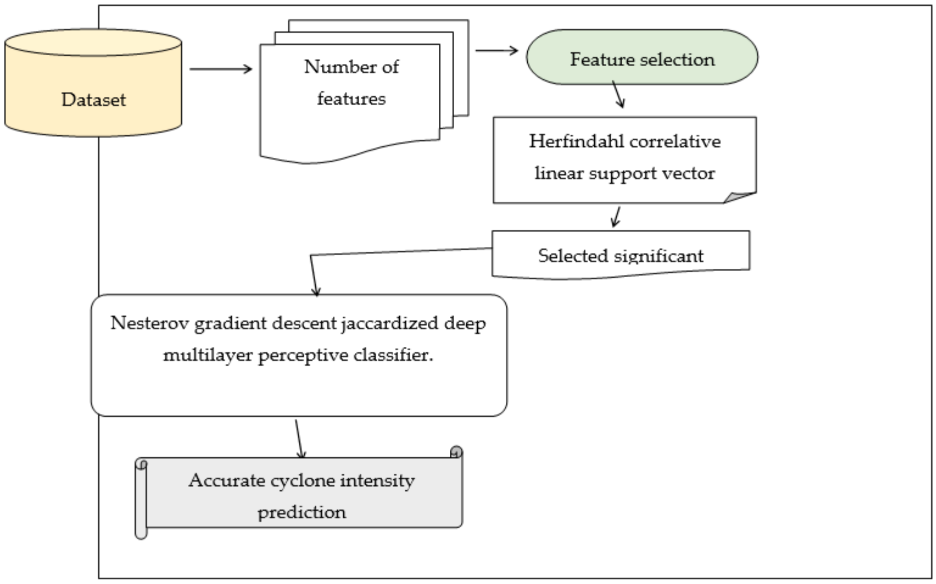

Figure 1. Architectural diagram of proposed LEGEMP technique

To decrease the time complexity of data categorization, the feature selection procedure must be carried out initially to discover which features are significant. The suggested approach selects significant features using a Herfindahl correlative linear support vector regression and excludes irrelevant features to improve the performance of cyclone intensity prediction in less time. Herfindahl correlative linear support vector regression is a machine-learning approach that helps identify the essential characteristics to improve cyclone intensity prediction while requiring less processing time.

Figure 1 shows a Nesterov gradient descent-based deep multilayer perceptron with linear support vector correlation, applying feature selection and Jaccardization for exact cyclone intensity prediction using the Herfindahl index and correlative analysis.

Secondly, hybridization of the Nesterov gradient descent Jaccardized deep multilayer perceptive classifier is employed to enhance cyclone intensity prediction performance with minimum time by estimating the testing and training parameters using the Jaccard similarity coefficient. Then, the soft step activation is applied to analyze the relevant value and predict the various cyclone intensities: minimum pressure, depression, deep depression, cyclonic storm, severe cyclonic storm, severe cyclonic storm, and super cyclonic storm. Finally, the Nesterov gradient descent method is applied to minimize the error by updating the weights. These two functions of the proposed LEGEMP technique are expressed in the following subsections.

3.1. Herfindahl Correlative Linear Support Vector Regression-Based Feature Selection

Feature selection is the most important machine-learning model, which greatly impacts the production of the classification model. Further features are supplied to the learning algorithm for a whole classification task. However, it is frequently the case that common features are irrelevant or redundant to the learning task, which minimizes the algorithm's performance and leads to the problem of achieving better accuracy in the prediction process. As a result, it is essential to choose the relevant, necessary features and eradicate the unnecessary features from the dataset.

The proposed LEGEMP technique uses Herfindahl correlative linear support vector regression to find the relevant features. Support vector regression is a machine-learning technique for estimating the relationships with variables, i.e., features. It is linear regression used to find and closely fit the features according to an exact mathematical criterion.

Figure 2 illustrates the feature selection process for identifying the significant features with minimum time using Herfindahl correlative linear support vector regression. Initially, the number of features and data is taken from the dataset. Then, input is given to the linear support vector regression model to identify the data with selected features with the help of the Herfindahl correlation index. It is a statistical method used to find the correlation between the relevant and irrelevant features, as depicted in Figure 3.

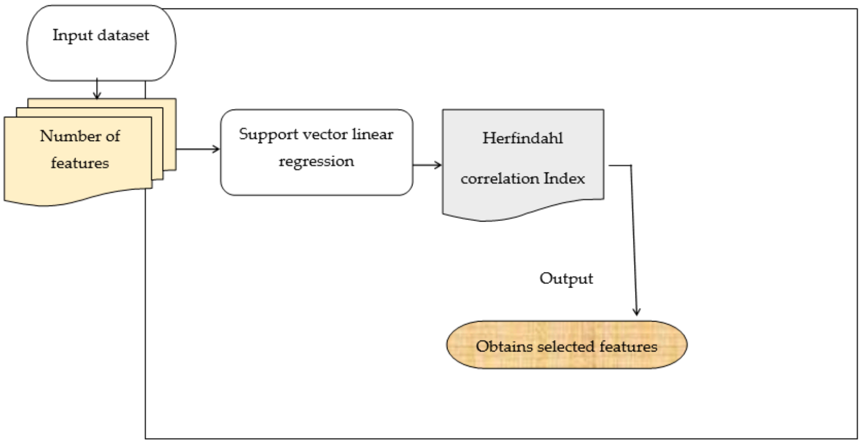

Figure 2. Herfindahl correlative linear support vector regression- based feature selection.

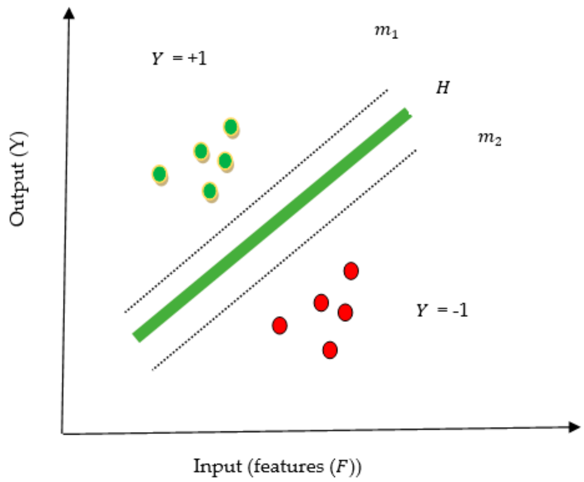

Figure 3. Process of Linear Support Vector Regression.

Initially, the features and the corresponding values were arranged in the form of a matrix with rows and columns, as given below:

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix} \tag{1}$$

From (1), a set of features, like relevant and irrelevant data, is represented as a matrix 'F' in which each column represents features and a row represents the feature data. In linear support vector regression, applying hyperplane is constructed for separating the original input vector space into two sets. After that, compute the relationship between features using the applied Herfindahl correlation function. The two marginal hyperplanes are constructed to split the input features into two sets according to the relationship between relevant and irrelevant features. All the subscripts of 'f' represent the rows and columns of features.

The regression model considers the set of training samples '(F, Y)', where 'F' represents the number of features matrix and 'Y' 'represents the output. The classification result provides two classes as $Y \in \{+1, -1\}$, where $Y = +1$ indicates the relevant features set and $Y = -1$ indicates the irrelevant features set. The support vector model uses the hyperplane to find the relevant or irrelevant feature set. The hyperplane is a decision boundary between the two sets, where the features are separated on either side of the decision boundary with the help of the Herfindahl correlation index. The Herfindahl correlation index that is used to calculate the relationship between features is measured as

$$HCI = \frac{1}{n} + n\left(\delta^2\right) \tag{2}$$

where HCI indicates a Herfindahl correlation index, 'n' indicates many features, and δ represents a statistical variance.

$$\delta^2 = \frac{1}{n-1}\sum\left|f_i - f_j\right|^2 \tag{3}$$

where $f_i,\ f_j$ denotes features. The correlation index provides the output value from 0 to 1. The hyperplane separates the relevant and irrelevant features above or below based on the correlation output. If the correlation results are higher, the hyperplane separates the relevant features above the decision boundary. If the correlation result is lesser, the hyperplane separates the irrelevant features below the decision boundary.

The output of support vector regression is given below:

$$Y = \begin{cases} +1; & relevant\ features \\ -1; & irrelevant\ features \end{cases} \tag{4}$$

In the above Equation (4), the positive results provide the maximum relationship between features, and the negative results present a lesser relationship between the features.

Figure 3 illustrates the yield of the linear support vector regression model. As represented in the figure, H denotes a hyperplane, i.e., decision boundary, and m_1 and m_2 denote a marginal between the two classes. These two margins are called the support vectors. The relevant features are separated into positive (i.e., +1), and irrelevant features are in negative classes (-1). The algorithm of the Herfindahl correlative linear support vector regression model is as follows. Algorithm 1 describes the different processes of data classification using the Herfindahl correlative linear support vector regression model for achieving better prediction outcomes with minimum time. As the input, the number of features and data is obtained. The original vector is then divided into two sets using the separating hyperplane in the regression model. The features are separated on either side of the hyperplane at every instant.

The Herfindahl correlation index is used to estimate the connection between the characteristics. When the link between the data is the strongest, the input characteristics are classified above the hyperplane. Otherwise, the input characteristics are grouped beneath the hyperplane. As a result, the necessary characteristics are identified, reducing the time complexity.

3.2. Nesterov Gradient Descent Jaccardized Deep Multilayer Perceptive Classifier for Cyclone Intensity Prediction

The second process of the proposed LEGEMP technique is to carry out the cyclone intensity prediction with selected features. The proposed technique uses the Nesterov gradient descent Jaccardized deep multilayer sensitive classifier with selected features. A deep multilayer perceptive classifier is a fully connected, feed-forward ANN that transforms any input dimension to the desired output. The multilayer deep perceptive classifier's main advantage over the other deep-learning techniques is to resolve complex nonlinear issues. It also handles huge amounts of input data and provides accurate predictions with less time after training.

In Algorithm 1, Herfindahl correlative linear support vector regression employs a linear approach within the framework of the Herfindahl index, enhancing regression accuracy by capturing market concentration. It optimally balances support vectors to model correlations, aiding effective predictions in economic analysis.

// Algorithm 1: Herfindahl correlative linear support vector regression

Input: cyclone dataset, number of features $f_1, f_2, f_3, \ldots, f_n$
Output: select significant features

Begin
  For each feature of $\boldsymbol{f_i}$ from the dataset
    Construct feature matrix '***F***'
    Construct hyperplane ***H***
    Find two marginal hyperplanes $\boldsymbol{m_1}$, $\boldsymbol{m_2}$
    Measure the Herfindahl correlation index '***HCI***'
    If the relationship between (f_i, f_j) is high
        $\boldsymbol{Y = +1}$
  the feature is said to be a relevant feature
  else
  $\boldsymbol{Y = -1}$
  the feature is said to be an irrelevant feature
  End if
  End for
  Return (relevant features)
End

Explanation of the Algorithm 1

1. A loop iterates over each feature in the dataset, denoted as f_i.
2. A matrix F is created that contains the dataset's features. Each row of F represents an instance, and each column represents a feature.
3. A hyperplane is established in the feature space, a subspace with one dimension less than the ambient space, used for classification or regression.
4. Two marginal hyperplanes (m_1 and m_2) are identified, potentially significant for further analysis or classification.
5. The Herfindahl correlation index (HCI), a measure quantifying the correlation between features, is calculated to assess feature distribution concentration or dispersion.
6. A conditional statement checks the HCI to evaluate the relationship between features f_i and f_j.
7. If HCI indicates a high relationship between features f_i and f_j, the variable Y is set to +1, marking f_i as relevant.
8. If the relationship between features f_i and f_j is low (i.e., low correlation), the variable Y is set to -1, indicating f_i as irrelevant.
9. The algorithm concludes with the end of the if–else statement, returning a list of relevant features based on the assessments made during the loop.

The perceptrons can process the given weighted sum of inputs, apply the activation function, and provide the final classification output. The relationship between perceptrons or neurons is known as the synapse. The perceptrons are a simplified computational model inspired by neurons, aiming to mimic certain aspects of neural behavior for specific computational tasks. In contrast, neurons are highly complex and specialized cells in biological nervous systems with many functions and capabilities.

Figure 4 illustrates the structural design of a deep multilayer perceptive learning classifier that includes an input layer, more than one output layer, and one hidden layer. The input layer receives the selected features input features of the cyclone with their data.

Each datum is connected with a set of weights '$q_1, q_2, \ldots, q_n$' and added with bias 'k'. The activity of the neuron is formulated as given below:

$$A(t) = \left[\sum_{i=1}^{n} B_i(t) * q_i\right] + h \tag{5}$$

where '$A(t)$' specifies a neuron's action at the input layer that weighted 'q_i' sum of the input data 'B' and adds to bias function 'h' stored, the value is '1'. Then, the input is transferred into the first hidden layer, where the deep-learning process is performed. More than one hidden layer is presented in the input and output layers. The hidden layer involves the small individual units called neurons or nodes. The motion of artificial neurons at the hidden layer is illustrated in Figure 3.

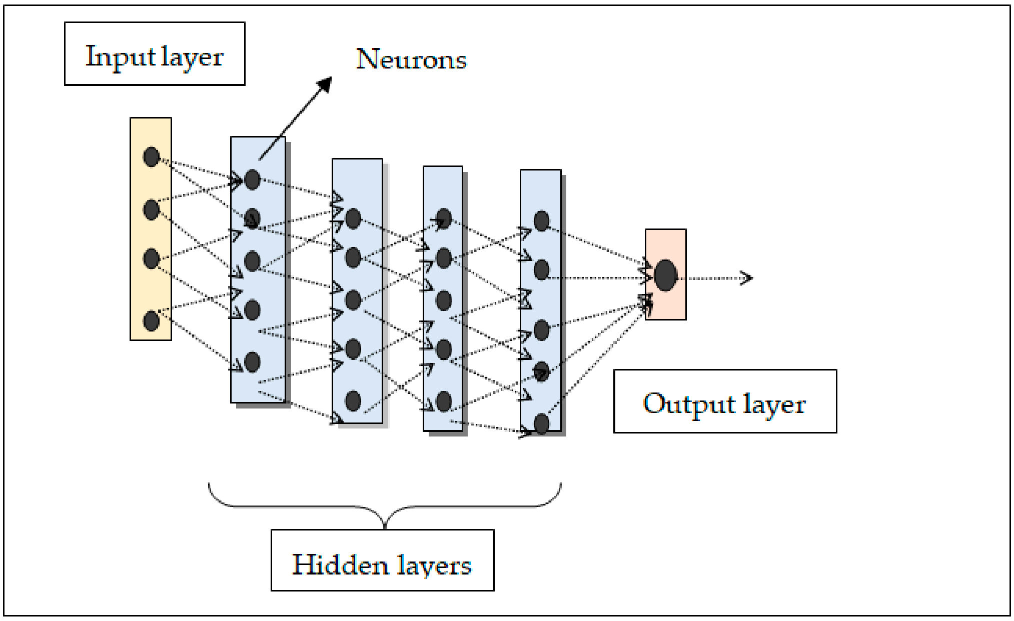

Figure 4. Structural design of deep multilayer perceptive learning classifier.

Figure 5 illustrates a process of artificial neuron activity that receives the weighted sum of features added with the bias. Then, the input data were analyzed using the cyclone's testing parameters and the Jacquard similarity method. The Jacquard coefficient is a statistical method used to compute the similarity between two sets of data, namely training data with testing parameters of the cyclone. The similarity is estimated as given below:

$$S = \frac{[Tr_B \cap Ts_B]}{\sum Tr_B + \sum Ts_B - [Tr_B \cap Ts_B]} \tag{6}$$

where S denotes a similarity coefficient, Tr_B represents the training parameters of the cyclone, Ts_B represents the testing parameters value of the cyclone, and $Tr_B \cap Ts_B$ denotes a mutual dependence between the data.

The coefficient (S) returns the output value from 0 to 1. The similarity outcomes are given to the input of the activation function for analyzing the value and providing the different intensity levels of the cyclone. The proposed deep-learning classifier uses the soft step activation role for providing final intensity prediction results.

The soft step activation metrics assist the network in learning complex testing and training patterns in data. The soft step activation function provides the best-normalized output between 1 and 0. It makes an accurate prediction. It also provides a bounded absolute value.

$$\varphi = [1 + \exp(-S)]^{-1} \tag{7}$$

where φ represents a soft step activation, and 'S' indicates the similarity outcomes. The soft step activation function provides the final classification results.

$$\varphi = \begin{cases} 1; & \textit{cyclone intensity level predicted} \\ 0; & \textit{otherwise} \end{cases} \tag{8}$$

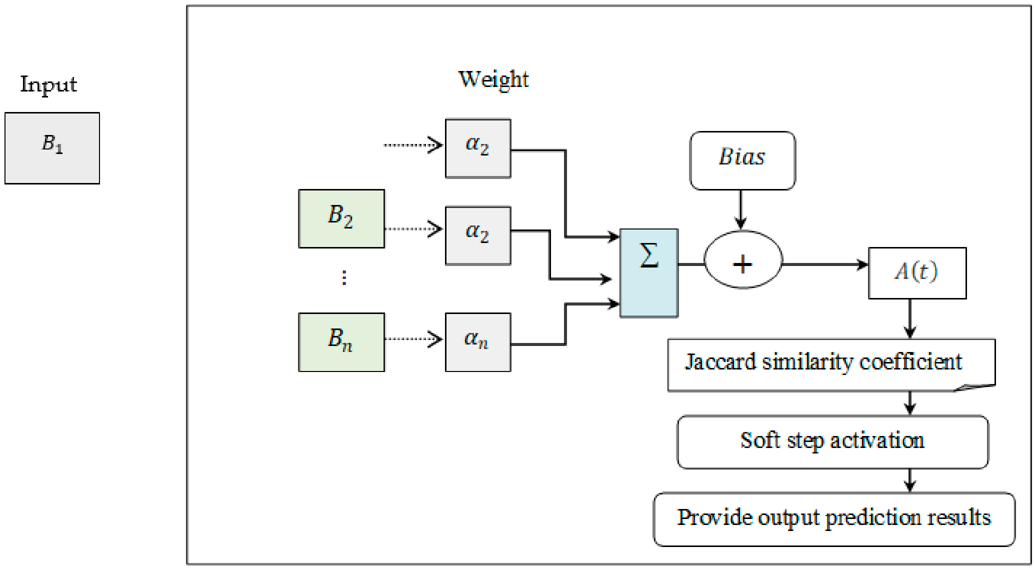

Figure 5. Process of artificial neuron activity.

The soft step activation provides '1', indicating that the cyclone intensity level is correctly predicted based on the similarity between testing and training parameter values. In the learning process in perceptron, the error rate is measured between the predicted results and the target output. The error rate is calculated as follows:

$$err = \frac{1}{2}\left[Z_t - Z_{pr}\right]^2 \tag{9}$$

where err represents the error rate, Z_t indicates the target prediction outcomes, and 'Z_{pr}' denotes a predicted cyclone intensity output produced with the deep multilayer perceptive classifier. The weight is updated using the Nesterov accelerated gradient descent method to minimize the error.

$$q_{t+1} = q_t - \eta v_t \tag{10}$$

where

$$v_t = \delta v_{t-1} + (1-\delta)\frac{\partial err}{\partial q_t} \tag{11}$$

where q_{t+1} is an updated weight, q_t is a current weight, and η denotes a learning rate $(\eta < 1)$. A huge learning rate permits the deep classifier to learn faster than a smaller value; '$\frac{\partial err}{\partial q_t}$' is a partial derivative of the error 'err' concerning current weight. 'q_t', δ default value is 0.9, and v_t is initialized at 0. The technique is repeated until the smallest error is found. Finally, the output layer of the multilayer perceptive classifier produces correct prediction results.

The different processes of cyclone intensity prediction with better accuracy and less time. Nesterov gradient descent using Jaccardization, a deep multilayer perceptive classifier, uses a variety of layers to assess the input data. The deep-learning classifier's input layer is supplied with the chosen pertinent features and their corresponding data. Then, the weight is assigned to each input and added with the bias.

In Algorithm 2, Nesterov gradient descent refines the deep multilayer perceptive classifier for cyclone intensity prediction, integrating Jaccard similarity. This augments accuracy by adapting learning rates, ensuring efficient convergence, and refining the forecasting of cyclone strength, using advanced neural network architectures.

// Algorithm 2: Nesterov gradient descent Jaccardized deep multilayer perceptive classifier-based cyclone intensity prediction

Input: Selected relevant features $\boldsymbol{F} = f_1, f_2, \ldots, f_k$ with the data $\boldsymbol{B} = B_1, B_2, \ldots, B_n$
Output: Improve the cyclone intensity prediction
Begin
Number of selected features $\boldsymbol{F} = f_1, f_2, \ldots, f_k$ with the data $\boldsymbol{B} = B_1, B_2, \ldots, B_n$ **are collected at the input layer**
For each datum '$\boldsymbol{B}$'
Assign weight '$\boldsymbol{q_i}$' and add bias '$\boldsymbol{h}$'
Obtain the neuron activity at the input layer '$\boldsymbol{A(t)}$'
End for
For each training data with testing parameter data **–[hidden layer]**
Measure the Jaccard similarity coefficient '$\boldsymbol{S}$'
Apply soft step activation function '$\boldsymbol{\varphi}$'
If $(\varphi = +1)$ **then**
Correctly predict the type of cyclone intensity
End if
End for
For each prediction outcome
Measure the error rate '***err***'
Apply the Nesterov accelerated gradient descent method
Update the weight '$\boldsymbol{q_{t+1}}$'
Find minimum error by identifying the optimal weight
Obtain the final prediction results with minimum error **at the output layer**
Return (accurate cyclone intensity prediction)
End for
End

The subsequently hidden layer neuron receives the weighted input after that. The Jacquard index is applied to compute the similarity between the input data and the testing cyclone parameter value in that layer. Then, the estimated similarity value is given to the soft step activation function at the hidden layer. The activation function evaluates the similarity value and outputs either a '1' or a '0'. The cyclone intensity level is correctly predicted if the output is '1'. Otherwise, the activation function returns '0'. The error rate is based on the squared difference between the target and the predicted output for each result. The initial weight is updated by applying a Nesterov accelerated gradient descent method to minimize the error. This process is continuously iterated until the algorithm reaches minimum error. Finally, the accurate cyclone intensity prediction results are displayed at the output layer.

4. Experimental Settings

Tong et al. and Na et al. used Python programming language to build regionally tailored meteorological datasets across the Bay of Bengal (BOB) to determine the strength of tropical cyclones in the region. The dataset comprised 72 cases, each with 15 different characteristics. They conducted a total of seven iterations, selecting instances in increments of 10 from 10 to 72. The 15 chosen features included information like cyclone duration, distance from the nearest seashore, average wind speed in 6 h intervals, central pressure, pressure drop in 6 h intervals, sea surface temperature (SST), cyclone height, eye diameter, cyclone orbit, translation speed, and landfall. These features were used as inputs for their approach. The last feature, wind speed or grade, was used to predict the severity of the cyclone.

5. Performance Results and Discussion

Researchers employed several metrics to evaluate the performance of the LEGEMP and ConvLSTM techniques developed by Tong et al. and Na et al. They investigated the prediction accuracy and error rate in predicting wind speed or gradient. They also assessed the time it took to make the predictions and the complexity of the necessary computing space.

5.1. Wind Speed/Grade Prediction Accuracy

These wind speeds are identified from the overall instances taken as input from the dataset. The accuracy is mathematically formulated as

$$WS/G_{acc} = \sum_{i=1}^{n} \frac{I_{AP}}{I_i} * 100 \tag{12}$$

where WS/G_{acc} denotes a wind speed/grade prediction accuracy that is calculated based on the number of instances involved in the simulation process. 'I_i' and I_{AP} denote the number of instances accurately predicted as wind speed. It is measured in percentage (%). The experimental results measured regarding wind speed/grade prediction accuracy are shown in the table.

Table 1 shows the feature descriptions with a clear brief, and Table 2 shows the cyclone intensity grades categorized by wind speed and potential damage. Ranging from Category 1 with minimal impact to Category 5 causing catastrophic destruction, these levels assess storm ferocity. The attributes include wind speed, storm surge, and atmospheric pressure, providing a vital gauge for forecasting and preparedness.

Table 1. Feature description.

S.No	Features	Description
1	Rec –id	Record identification (Basin of origin)
2	life of cyclone (Days)	Duration of cyclone in days
3	Distance nearest sea shore (km)	A land located from the body of water in kilometers
4	Max Wind Speed (Knot)	Maximum wind speed
5	Avg. Wind speed (6 h int)	Average wind speed
6	Central pressure (Mb)	Average pressure at the center of the surface in millibars
7	Avg. pressure drop (6 h)	Average pressure drop
8	SST (Celsius)	Sea surface temperature
9	Cyclone Diameter	Diameter of cyclone
10	Cyclone Height	Height of cyclone above the sea surface
11	Eye Diameter	Average size (diameter) of the eye of a cyclone
12	TC_Orbit (Km)	Curved trajectory of a tropical cyclone (TC)
13	Translation Speed (KMPH)	Translation Speed of a hurricane in kilometers per hour
14	Land Fall	Event of a cyclone moving over land after being over water

Table 2. The cyclone intensity level in grade attributes.

Wind Speed or Grade	Description
D	Depression
DD	Deep Depression
CS	Cyclonic storm
SCS	Severe cyclonic storm
VSCS	Very severe cyclonic storm
ESCS	Extremely severe cyclonic storm

Table 3 shows the accuracy of wind speed/grade prediction versus the number of instances gathered from the Bay of Bengal dataset. The data instances are gathered from the Bay of Bengal origin. Each instance comprises 11 parameters or attributes or features, such as the life of the cyclone, distance nearest seashore, average wind speed (6 h interval), central pressure, average pressure drop (6 h interval), sea surface temperature (SST), cyclone height, eye diameter, TC_Orbit, translation speed, and landfall. Figure 6 shows the number of instances in the x-axis and their corresponding wind speed or grade prediction accuracy rate in the y-axis. From the above graphical illustration, the accuracy of the three different methods is decreased while increasing the number of instances. The simulations calculate the wind speed/grade prediction accuracy for ten instances. From the instances, nine correctly predicted the wind speed/grade using the LEGEMP method, eight correctly predicted using Tong et al. [1], and seven instances were correctly predicted using Na et al. [1].

Table 3. Comparison of Wind speed/grade prediction accuracy.

Number of Instances	Wind Speed/Grade Prediction Accuracy (%)		
	LEGEMP	ConvLSTM	RCTCP
10	90	80	70
20	90	80	70
30	86.66	76.66	70
40	85	75	67.5
50	84	74	66
60	83.33	73.33	65
72	82.85	72.85	64.28

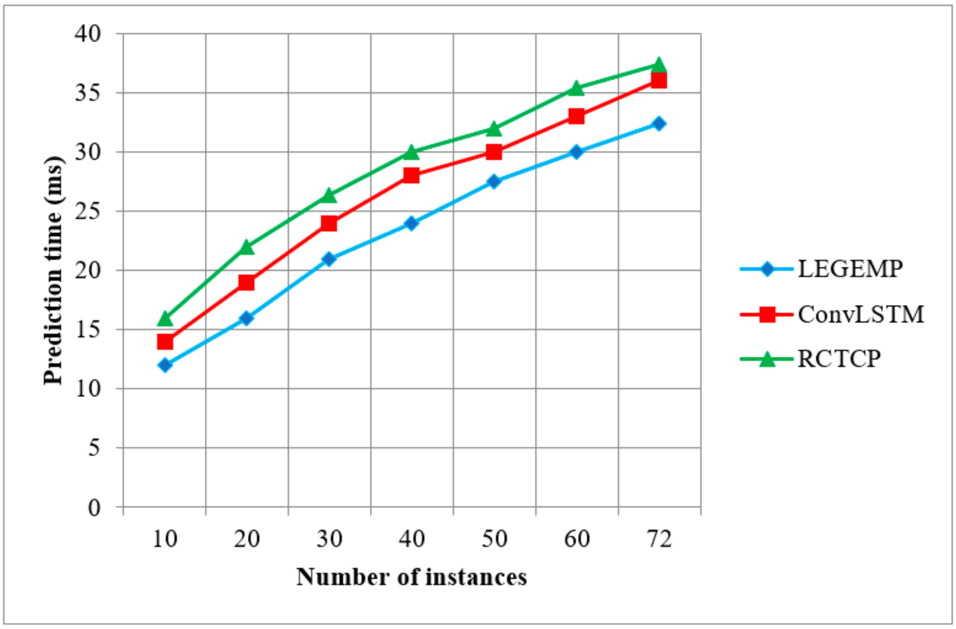

Figure 6. Graphical illustration of prediction time.

As a result, the overall wind speed/grade prediction accuracy using the three methods was observed to be 90%, 80%, and 70%, respectively. In the observed results, wind speed/grade prediction accuracy using the LEGEMP method is comparatively improved compared to that of Tong et al. [1] and Na et al. [2]. The motivation after the improvement was due to the application of the Nesterov gradient descent Jaccardized deep multilayer perceptive classifier. Applying the proposed deep-learning method, the similarity between the training data, such as the 11 parameters and their testing values, is estimated using the Jacquard index coefficient. The estimated similarity value is given to the soft step activation function. The activation function evaluates the similarity value and delivers the results based on how accurately the wind speed or grade predictions were made. For each iteration, the proposed technique is compared to the conventional methods. Then, the average of seven values indicates that LEGEMP enhances the wind speed/grade prediction accuracy by 13% and 27% compared to that of ConvLSTM [1] and [2], respectively.

Table 4 presents the results of employing the different models, including LEGEMP, ConvLSTM, and RCTCP, to evaluate the accuracy, precision, recall, and F1-score. These measures are essential in determining the effectiveness of the proposed model and ensuring that it outperforms the other models. They provide a finer-grained idea of how well a classifier is doing instead of just looking at overall accuracy. The F1 score is particularly useful as it balances precision and recall, ensuring the model is not biased towards either metric.

Table 4. Models with metric accuracy in percentage.

Models	Accuracy %	Precision %	Recall %	F1-Score %
LEGEMP	82.85	81.71	80.57	79.43
ConvLSTM	72.85	71.64	70.43	69.22
RCTCP	64.28	63.41	62.54	61.67

5.2. Error Rate

The error rate refers to the percentage ratio of several instances to predict the overall sample data taken as input incorrectly. The error rate is mathematically formulated as

$$P_{Err} = \sum_{i=1}^{n} \frac{IIP}{I_i} * 100 \quad (13)$$

where the 'P_{Err}' wind speed/grade prediction error rate is estimated based on instances involved in the simulation process. 'I_i' and instances are incorrectly predicted 'IIP'. It is measured in percentage (%). The experimental results measured in error rate are shown in Table 5.

Table 5. Comparison of error rate.

Number of Instances	Error Rate (%)		
	LEGEMP	ConvLSTM	RCTCP
10	10	20	30
20	10	20	30
30	13.33	23.33	30
40	15	25	32.5
50	16	26	34
60	16.66	26.66	35
72	18.05	27.77	36.11

Table 5 depicts the error rate using the number of instances. Also, from the above figure, the LEGEMP method's error rate is minimized more than the other two existing methods. The experiments are performed with seven instances for calculating the wind speed or grade prediction error. Let us consider ten instances in the first iteration. As a result, the prediction error rate using the three methods, LEGEMP, Tong et al. 2022 [1], and Na et al. 2022 [2], was observed to be 10%, 20%, and 30%, respectively. From the analysis, the LEGEMP outperforms well, minimizing the error rate for wind speed or grade prediction. The reason behind the error rate minimization was the application of a Nesterov accelerated gradient descent method in the proposed deep- learning classifier. The method minimizes the error rate of intensity prediction by updating the weight of samples. This process is continuously iterated until the algorithm reaches minimum error. The average error rate for seven iterations is minimized by 42% and 57% compared to Tong et al. [1] and Na et al. [2].

5.3. Prediction Time

This measure is important since it allows for early prediction. It is described as the time an algorithm requires to estimate wind speed or grade. Therefore, the overall prediction time is mathematically stated as

$$P_T = \sum_{i=1}^{n} I_i * Time\ [IP] \tag{14}$$

From the above Equation (14), 'P_T' is the prediction time measured based on the number of instances involved in the simulation process and time consumed for one instance in actual wind speed or grade prediction '$Time\ [IP]$'.

It is measured in milliseconds (ms). The experimental results measured in cyclone intensity prediction time are illustrated. In Figure 6, the graphical representation of prediction time is shown. The figure shows that the 'y' axis represents the wind speed or grade prediction time, and 'the x-axis represents the number of instances that involve the eleven parameters, such as the life of the cyclone, distance nearest seashore, average wind speed (6 h interval), central pressure, average pressure drop (6 h interval), sea surface temperature, cyclone height, eye diameter, TC_orbit, translation speed, and landfall. From the above graph, using all three methods, LEGEMP, ConvLSTM [1], and [2], the performance of the prediction time depends on the number of instances available in the dataset. However, the experiments were conducted with ten instances for calculating the prediction time. By applying LEGEMP, 12 ms as been consumed for wind speed or grade prediction.

Similarly, 14 ms and 16 ms were considered intensity predictions using Tong et al. [1] and Na et al. [2], respectively. As a result, the prediction time using LEGEMP was comparatively reduced to that of Tong et al. [1] and Na et al. [2]. The after motive for the development was the purpose of Herfindahl correlative linear support vector regression. This regression method uses the hyperplane to split the features into two parts based on the Herfindahl correlation function. Based on the correlation outcomes, relevant features are identified and other features. The selected features perform wind speed or grade prediction, reducing time consumption.

5.4. Space Complexity

Space complexity refers to the amount of space the algorithm consumes for accurate wind speed or grade prediction. Space complexity is mathematically stated as given below:

$$SC = \sum_{i=1}^{n} I_i * Mem\ [IP] \tag{15}$$

From the above Equation (15), 'SC' is the space complexity measured based on the number of instances 'I_i' involved in the simulation process and memory consumed during wind speed or grade prediction '$Mem\ [IP]$'. It is measured in kilobytes (KB). The experimental results measured in terms of space complexity are described in Table 6.

Table 6. Comparison of space complexity.

Number of Instances	Space Complexity (KB)		
	LEGEMP	**ConvLSTM**	**RCTCP**
10	16	19	21
20	22	26	28
30	31.5	36	39
40	36	40	44
50	39	42	46
60	41.4	45	48
72	43.2	50.4	52.56

Finally, Table 6 above illustrates the spatial complexity involved in predicting wind speed or grade. The performance results of space complexity are discovered to be lower when employing LEGEMP than when using the existing approaches. However, experiments are carried out with 10 instances to calculate the space complexity. As a consequence, the space complexity of the LEGEMP approach was lower than that of ConvLSTM Tong et al. [1] and Na et al. [2]. The feature selection technique for cyclone intensity prediction is applied as an after-reason enhancement. Using this technique, relevant feature samples may be favored, while unnecessary characteristics or parameters are removed. This helps to reduce the memory use of wind speed or grade forecast. The average of seven findings shows that the space complexity is reduced by 12% and 18%, respectively, compared to that of the current Tong et al. [1] and Na et al. [2].

6. Conclusions

This study introduces significant advancements, notably the development of LEGEMP, to enhance model accuracy. Additionally, it utilizes Herfindahl correlative linear support vector regression to minimize prediction time for cyclone intensity. Furthermore, feature selection is optimized using the Nesterov gradient descent Jaccardized deep multilayer perceptive classifier. The development of the LEGEMP, incorporating Herfindahl correlative linear support vector regression for effective feature partitioning and selection, has shown promising advancements in cyclone intensity level forecasting. The application of the Nesterov gradient descent Jaccardized deep multilayer perceptive classifier, aided by the Jaccard similarity index, has significantly improved the accuracy and efficiency of feature selection. Utilizing the soft step activation function for output generation has further enhanced the reliability of intensity predictions. Employing the Nesterov gradient descent approach for weight updates has effectively minimized cyclone intensity forecast error rates. These combined strategies demonstrate the potential of LEGEMP in optimizing cyclone intensity prediction, ultimately contributing to more precise and timely forecasting, which is crucial for disaster preparedness and mitigation. Further research and real-world validation are warranted to validate and fine-tune the effectiveness and robustness of LEGEMP in cyclone intensity forecasting.

Author Contributions: Conceptualization, S.K.J.; methodology, V.V.; software, M.M.E. draft preparation, K.K.; writing—review and editing, S.K.R.; visualization, D.S.K.; data curation, F.K.K.; and resources, A.E.A. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of the INSAT3D's INFRARED and RAW cyclone images from 2012 to 2021 over the Indian Ocean are included in this image dataset along with each cyclone's strength in knots. The MOSDAC server was used to obtain the raw data, and I named each image by finding the date and corresponding position in the intensity–time graph of each cyclone directory. Dataset Kaggle URL: https://www.kaggle.com/datasets/sshubam/insat3d-infrared-raw-cyclone-images-20132021. Willing to take full responsibility for the article, including for the accuracy and appropriateness of the reference list. Data accessed on 30 August 2023.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: No conflicts raised by the authors.

References

1. Tong, B.; Wang, X.; Fu, J.; Chan, P.; He, Y. Short-term prediction of the intensity and track of tropical cyclone via ConvLSTM model. *J. Wind. Eng. Ind. Aerodyn.* **2022**, *226*, 105026. [CrossRef]
2. Abed, M.; Alam Imteaz, M.; Ahmed, A.N.; Huang, Y.F. Modelling monthly pan evaporation utilising Random Forest and deep learning algorithms. *Sci. Rep.* **2022**, *12*, 13132. [CrossRef] [PubMed]
3. Oad, S.; Alam Imteaz, M.; Mekanik, F. Artificial Neural Network (ANN)-Based Long-Term Streamflow Forecasting Models Using Climate Indices for Three Tributaries of Goulburn River, Australia. *Climate* **2023**, *11*, 152. [CrossRef]
4. Ghamariadyan, M.; Imteaz, M.A. Prediction of seasonal rainfall with one-year lead time using climate indices: A Wavelet Neural Network scheme. *Water Resour. Manag.* **2021**, *35*, 5347–5365. [CrossRef]
5. Chan, M.H.; Wong, W.K.; Au-Yeung, K.C. Machine learning in calibrating tropical cyclone intensity forecast of ECMWF EPS. *Meteorol. Appl.* **2021**, *28*, e2041. [CrossRef]
6. Lee, Y.-J.; Hall, D.; Liu, Q.; Liao, W.-W.; Huang, M.-C. Interpretable tropical cyclone intensity estimation using Dvorak-inspired machine learning techniques. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104233. [CrossRef]
7. Wang, X.; Wang, W.; Yan, B. Tropical Cyclone Intensity Change Prediction Based on Surrounding Environmental Conditions with Deep Learning. *Water* **2020**, *12*, 2685. [CrossRef]
8. Xu, W.; Balaguru, K.; August, A.; Lalo, N.; Hodas, N.; DeMaria, M.; Judi, D. Deep learning experiments for tropical cyclone intensity forecasts. *Weather. Forecast.* **2021**, *36*, 1453–1470. [CrossRef]
9. Giffard-Roisin, S.; Yang, M.; Charpiat, G.; Bonfanti, C.K.; Kégl, B.; Monteleoni, C. Tropical Cyclone Track Forecasting Using Fused Deep Learning from Aligned Reanalysis Data. *Front. Big Data* **2020**, *3*, 1. [CrossRef]
10. Jiang, S.; Fan, H.; Wang, C. Improvement of Typhoon Intensity Forecasting by Using a Novel Spatio-Temporal Deep Learning Model. *Remote. Sens.* **2022**, *14*, 5205. [CrossRef]
11. Lian, J.; Dong, P.; Zhang, Y.; Pan, J.; Liu, K. A Novel Data-Driven Tropical Cyclone Track Prediction Model Based on CNN and GRU With Multi-Dimensional Feature Selection. *IEEE Access* **2020**, *8*, 97114–97128. [CrossRef]
12. Higa, M.; Tanahara, S.; Adachi, Y.; Ishiki, N.; Nakama, S.; Yamada, H.; Ito, K.; Kitamoto, A.; Miyata, R. Domain knowledge integration into deep learning for typhoon intensity classification. *Sci. Rep.* **2021**, *11*, 1–10. [CrossRef] [PubMed]
13. Zhou, Y.; Zhao, J.; Zhan, R.; Chen, P.; Wu, Z.; Wang, L. A Logistic-growth-equation-based Intensity Prediction Scheme for Western North Pacific Tropical Cyclones. *Adv. Atmos. Sci.* **2021**, *38*, 1750–1762. [CrossRef]
14. Sridevi, C.; Pattanaik, D.; Das, A.; Srivastava, A.; Durai, V.; Johny, C.; Deshpande, M.; Suneetha, P.; Kanase, R. Tropical cyclone track and intensity prediction skill of GFS model over NIO during 2019 & 2020. *Trop. Cyclone Res. Rev.* **2022**, *11*, 36–49. [CrossRef]
15. Sarkar, I.; Chaudhuri, S.; Pal, J. Forecasting pressure drop and maximum sustained wind speed associated with cyclonic systems over Bay of Bengal with neuro-computing. *Theor. Appl. Clim.* **2022**, *149*, 1255–1276. [CrossRef]
16. Wahiduzzaman; Cheung, K.K.; Luo, J.-J.; Bhaskaran, P.K. A spatial model for predicting North Indian Ocean tropical cyclone intensity: Role of sea surface temperature and tropical cyclone heat potential. *Weather. Clim. Extremes* **2022**, *36*, 100431. [CrossRef]
17. Singh, K.S.; Nayak, S.; Maity, S.; Nayak, H.P.; Dutta, S. Prediction of Bay of Bengal Extremely Severe Cyclonic Storm" Fani" Using Moving Nested Domain. *Preprints* **2022**, 1–14. [CrossRef]
18. Tian, W.; Huang, W.; Yi, L.; Wu, L.; Wang, C. A CNN-Based Hybrid Model for Tropical Cyclone Intensity Estimation in Meteorological Industry. *IEEE Access* **2020**, *8*, 59158–59168. [CrossRef]
19. Devaraj, J.; Ganesan, S.; Elavarasan, R.M.; Subramaniam, U. A Novel Deep Learning Based Model for Tropical Intensity Estimation and Post-Disaster Management of Hurricanes. *Appl. Sci.* **2021**, *11*, 4129. [CrossRef]
20. Maskey, M.; Ramachandran, R.; Ramasubramanian, M.; Gurung, I.; Freitag, B.; Kaulfus, A.; Bollinger, D.; Cecil, D.J.; Miller, J. Deepti: Deep-Learning-Based Tropical Cyclone Intensity Estimation System. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2020**, *13*, 4271–4281. [CrossRef]
21. Alves, R.G.; Maia, R.F.; Lima, F. Development of a Digital Twin for smart farming: Irrigation management system for water saving. *J. Clean. Prod.* **2023**, *388*, 135920. [CrossRef]
22. Obaideen, K.; Yousef, B.A.; AlMallahi, M.N.; Tan, Y.C.; Mahmoud, M.; Jaber, H.; Ramadan, M. An overview of smart irrigation systems using. *I.O.T. Energy Nexus* **2022**, *7*, 100124. [CrossRef]
23. Kurtulmuş, E.; Arslan, B.; Kurtulmuş, F. Deep learning for proximal soil sensor development towards smart irrigation. *Expert Syst. Appl.* **2022**, *198*, 16812. [CrossRef]