Article

Advanced Weather Prediction based on Hybrid Deep Gated Tobler's Hiking Neural Network and robust Feature Selection for tackling Environmental Challenges

Jasmine J[1], Sonia Jenifer Rayen[2], Ramesh T[3], Surendran Rajendran[4]

[1] Department of Computer Science and Engineering, Karpagam College of Engineering, Coimbatore 641032, India. jas378@gmail.com

[2] Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai 600119, India. sonijr1@gmail.com

[3] Department of Computer Science and Engineering, R.M.K Engineering College, Kavaraipettai, Chennai 601206, India. trh.cse@rmkec.ac.in

[4] Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India. surendran.phd.it@gmail.com

*Corresponding author: Surendran Rajendran

E-mail: surendran.phd.it@gmail.com

ABSTRACT

Human activities are directly affected by weather events. In particular, extreme weather events like forest fires, global warming, drought-causing high air temperatures make human life challenging. The use of reliable and accurate weather prediction models is essential to take precautions against these types of climate events. As a result, creating models that accurately forecast the weather is critical. The successful development of deep learning-based weather prediction models has largely aided by technological advancements. With high success rate, this paper proposes a Robust Feature Selection based Hybrid Weather Prediction (RFS-HWP) model for weather prediction. The input dataset is

initially pre-processed with the help of Missing Data Imputation and Z-score Standardization. After that, the feature selection process is accomplished using Botox Optimization Algorithm (BxOA) to find the optimal subset of features. The selected features are then fed into the Hybrid Deep Gated Tobler's Hiking Neural Network (HDGT-HNN) model, which classifies weather conditions into three classes as temperature, pressure and humidity. The hyper-parameters of HPC-DBCN are optimized using Hiking Optimization Algorithm (HiOA). The entire implementation is carried out on Python platform using publicly available Jena climate dataset, and many types of performance measures are calculated. Also, the usefulness of RFS-HWP model is proven by comparing its performance to state-of-the-art approaches. As a result, the RFS-HWP outperforms by accomplishing overall accuracy of 99.3% and proven to be an applicable model for weather forecasting systems.

Keywords: Weather prediction; Deep learning; Botox optimization; Feature selection; Hiking optimization; Gated recurrent unit; Attention mechanism;

1. Introduction

The ability to predict weather conditions, such as temperature and humidity, allows us to understand the climate and make decisions accordingly. Wind, humidity, temperature etc., are examples of environmental factors that influence human livelihood. Weather forecasting is termed as a scientific technique of predicting the atmospheric state based on particular time frames and locations [1, 2]. Climate disaster prediction and warning depend heavily on early environmental forecasting. A variety of societal advantages, comprising the preservation of life and property, the maintenance of economic growth and quality of life, and the advancement of public health and safety are supported by accurate weather prediction [3]. For these reasons, this field of early prediction is crucial. Farmers could benefit from having access to current weather reports in order to maximize productivity.

Predicting the course of weather is crucial in halting the spread of pests and diseases among crops. The weather may have an impact on pests that consume crops.

Numerical Weather Prediction (NWP) techniques are the basis of earlier works. It is associated to numerous difficulties, particularly a lack of understanding and knowledge of physical mechanisms [4-6]. The process of extracting relevant information from huge amounts of data necessitates high processing power and powerful computers. The application of physical model determines the accuracy of NWP technique. It is generally depending on nonlinear equations termed as primitive equations [7-8]. When the model receives input with inadequate knowledge about the atmosphere, the minor variations in the atmosphere will have more effect on the outcome of model. The error, which is the discrepancy between the actual and predicted times automatically decreases the accuracy. The deluge of information is yet another major issue with the existing methods. Weather forecasting methods based on deep learning (DL) have advanced dramatically in the recent decades [9].

Computer vision applications for DL technology are numerous, and time series issues can effectively forecast both temporal and spatial information [10-12]. Meteorological data becomes the sample and standard geographical data. An excellent alternative for traditional weather forecast method is the DL-based approach. By implementing the artificial system, DL models [13-16] like Artificial Neural Networks (ANN), which mimic the structure of human brain and relate the complex relationship between the task's output and input, are more dominant data modeling technology. DL techniques make it simple to address nonlinear problems, such as weather prediction. Recurrent Neural Networks (RNN) and Temporal Convolutional Networks (TCN) are popular deep models for multivariate time-series predicting [17, 18]. Due to its improved performance, an RNN version recognized as Long Short-Term Memory (LSTM) has drawn greater interest [19, 20].

Artificial Neural Networks (ANNs) represent a significant deep learning model which replicates human brain structures to find complex relationships in input-output interactions. Multiple domains use these models as their primary data modeling equipment. ANNs have successfully forecasted renewable energy while improving system optimization according to research publications [21, 22]. Electricity load forecasting together with grid stability analysis depends on ANNs to ensure efficient resource management in smart grids [23, 24]. Neural network developments have improved their capacity to work with dynamic and non-linear datasets which makes them essential for weather prediction and energy system management [25]. ANS became increasingly essential because of their status as a leading artificial intelligence technology that performs reliably across various applications.

These models' exceptional performance has gathered a lot of attention. Besides, the deep networks frequently employ stacked neural networks, with multiple layers making up the overall structure acknowledged as nodes. Since the node allows for the combining of data input over coefficient sets, the computation complexity of these model has maximized. Thereby, this paper focusses on an advanced weather prediction model using optimized deep learning and effective feature selection. With the support of the link between the selected features as well as their representation, the proposed model maximizes the accuracy and minimizes the complexity. The major contribution of the proposed work is presented below as follows:

- To design a Robust Feature Selection based Hybrid Weather Prediction (RFS-HWP) model for effectually predicting weather conditions like temperature, pressure and humidity.
- To select the optimal features and minimize the dimensionality issues, a new meta-heuristic optimization algorithm called Botox Optimization Algorithm (BxOA) is employed.

- To present a Hybrid Deep Gated Tobler's Hiking Neural Network (HDGT-HNN) model, which incorporates Additive Attention Mechanism (AAM) and Hiking Optimization Algorithm (HiOA) based hyperparameter tuning, for the prediction of weather with maximum precision rate.
- To estimate the performance of RFS-HWP model by relating it with prevailing models for determining the superiority and applicability in weather monitoring systems.

The paper is summarized as follows: Section 2 deliberates the recent work done by different authors related to weather prediction. Section 3 signifies the methodology of RFS-HWP model. Section 4 provides the experimental outcomes of RFS-HWP model. Finally, the conclusion and future work are presented in Section 5.

2. Related Works

A. Utku & U. Can [26] presented a hybrid weather forecasting approach using RNN and convolutional neural network (CNN) with maximum success rate. Here, the large-scale, 14-parameter meteorological data in the Jena dataset was subjected for weather prediction. To compare the experimental results, popular machine learning, deep learning, and statistical techniques were used and produced the best forecast outcome for all metrics. This hybrid method had obtained 0.126% Mean Absolute Error (MAE), 0.035% Mean Squared Error (MSE), 0.189% Root MSE (RMSE), and 0.987% R-Squared, respectively. However, this method undergoes vanishing and gradient explosion issues, and require more computation resources. Rita Teixeira et al. [27] presented a regression model based on LSTM to predict medium- to short-term weather in Douro region. The hyperparameters of LSTM were tuned by utilizing a genetic algorithm (GA). The outcome demonstrated that the optimized LSTM had minimized the evaluation criteria throughout a range of time horizons. This is because the LSTM utilized gates and memory cells to store previous information and update or forget it as new data were

presented. This allowed LSTMs to recognize and understand long-term dependencies within the data sequence. However, it became clear that the computing cost of training the models would be significant. Leopoldo Eduardo Cardenas-Barron and Alfonso Angel Medina-Santana [28] recommended an optimal design considering weather forecasting using RNN in order to forecast three variables such as wind speed, global solar irradiance, and ambient temperature. Here, the RMSprop optimizer was used by RNN to train the hyperparameters. Even with improved performance, other metaheuristic optimization strategy would be essential in order to tackle the optimization problem and reduce the complexity. Additionally, the inclusion of improved RRN networks for weather parameter prediction could be helpful. K. Venkatachalam et al. [29] suggested an enhanced data-driven hybrid model for predicting weather based on transductive LSTM (T-LSTM). The evaluation metrics such MAE, loss, and RMSE were used for evaluating the model. The Jena Climate and HHWD datasets were employed for the experiments. There were 14 weather forecasting features in the dataset, like temperature and humidity. The T-LSTM model had outperformed other approaches, and yielded 98.2% of accuracy rate in weather prediction. Subsequently, the hybrid T-LSTM model offered a reliable solution for hydrological variables. The drawbacks of this model would include overfitting problems, less robustness, and limited ability to unseen data.

Sercan Yalcın [30] introduced a deep hybrid neural networks for weather condition forecasting with time series. This technique utilized a hybrid approach of training and predicting weather forecasting parameters using LSTM and CNNs models. The parameters were relative humidity, pressure, temperature, and wind speed. First, the CNN layers processed the values resembled by the input neurons in order to deliver a more precise and understandable data assessment. Then, the outcomes were fed to the LSTM block after optimization. Based on the experimentation, the relative humidity,

temperature, atmospheric pressure, and wind speed RMSE values were 7.12, 1.82, 2.61, and 1.06, respectively. However, the model exhibited overfitting issues and higher error rate.

Problem statement: The increasing complexity of weather systems and the limitations of conventional NWP techniques, which often struggles with sensitivity and high computational demands, highlight the urgent requirement for more effective weather prediction methods. Earlier works have exposed the potential of DL methods, specifically CNN, RNN but they are prone to overfitting and tending to poor generalization on unseen data. Besides, the computational complexity allied with training DL can be crucial, requiring substantial processing time and power, which may not be possible for real-time applications. Furthermore, the dynamic and nonlinear nature of weather systems is often not satisfactorily taken into account by existing methodologies, which can tend to errors, particularly in situations where conditions are changing quickly. The LSTM model captures intricate spatial and temporal relationships in meteorological data. Yet, there remains a significant gap in the application of DL for weather prediction by using several meteorological parameters and optimizing the model parameters by parameter tuning. A significant number of existing methods have concentrated on limited parameter sets and not completely explored the advantages of hybrid models that incorporates various DL approaches. Thereby, an efficient and robust prediction method is needed to address the challenges of existing methods by associating various meteorological variables.

3. Proposed Methodology

In this section, an efficient RFS-HWP model, which utilizes weather data is discussed to predict patterns and forecasting with better precision. The steps involved in RFS-HWP framework are pre-processing, feature selection and deep-learning based weather prediction. Initially, the input data required for weather prediction process are collected from the Jena climate dataset. The pre-processing

steps are performed over the raw input data for generating the relevant outcomes valuable for an effective prediction. After pre-processing, effective features are extracted using Botox Optimization Algorithm (BxOA). After extracting the relevant features, the data are classified through proposed predictive model. The overall performance in the predictive network is optimized by tuning the hyperparameters. The RFS-HWP model helps to enhance the weather prediction process and serves as an excellent research source. The workflow of RFS-HWP architecture is described in Figure 1.

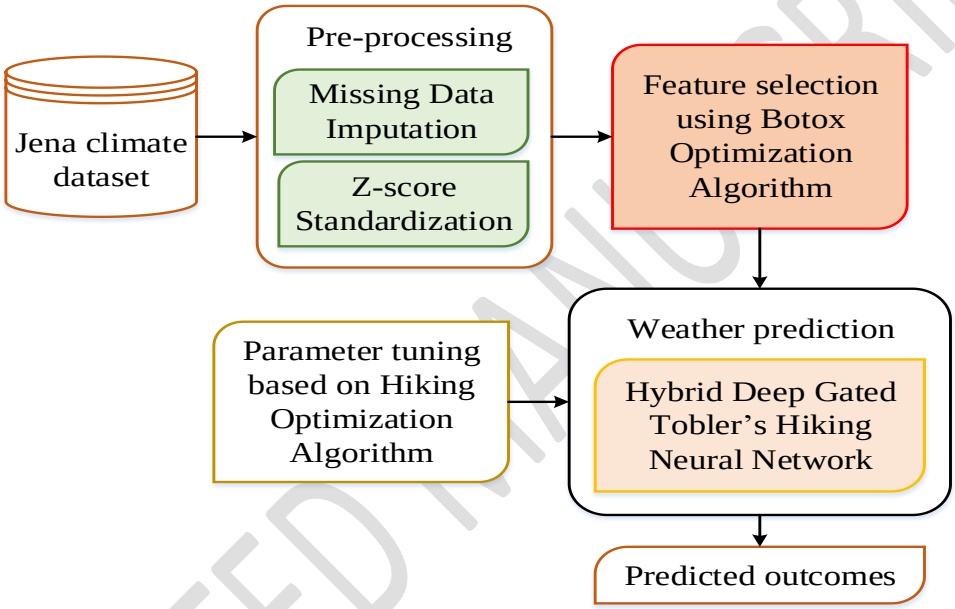


Figure 1: Block Diagram of RFS-HWP Model

3.1. Pre-Processing

A crucial phase in the pipeline for data analysis and deep learning is data pre-processing, which involves converting raw data from the dataset into a clean and readable format. This process includes enhancing the quality of data and ensuring it is appropriate for modeling or analysis. The design variables encompass the parameters for Missing Data Imputation and the thresholds for Z-score Standardization, which facilitate the dataset readiness for thorough analysis. In RFS-HWP model, the following are the major components of preprocessing the data. Missing value imputation: One of the biggest challenge to practical data collection is deliberated as missing data [31]. In data-driven model

construction, missing values tend to various difficulties such as inefficiencies and other complexity. It can result in biased decisions because complete and missing data differ from one another. As the missing data affects the model performance, a missing value imputation must be performed. Two kinds of missing data imputation techniques are usually employed in earlier works. The first approach is depending on a strategy of missing data tolerance that combines specific data mining algorithms with missing value techniques. On the other hand, the RFS-HWP model substitutes missing values with the median of feature's non-missing values by means of a standard imputation method with numerical features.

$$x(k) = x(Median) \quad (1)$$

where, suggests the missing values replaced with median of observed data, and defines the median value of observed data points.

Z-score normalization: When the underlying attribute range is unidentified or an outlier exists, min-max normalization can be strongly affected or not feasible [32]. Z-score normalization [33] is another method that involves transforming the data so that it has 1 as the standard deviation and 0 as the mean. Given the attribute's mean and standard deviation, the transformation is characterized as in below equation.

$$\delta' = \frac{\delta - \eta}{\rho} \quad (2)$$

where, represents the normalized value. It should be noted that the sample mean and standard deviation could be utilized if and are unknown.

3.2. Feature Selection

One of the most challenge in recent decades is choosing the appropriate features to solve the data classification issues with best outcome. The design variables represent the selected subset of

meteorological features, including temperature, pressure, and humidity, identified by the Botox Optimisation Algorithm (BxOA) to minimise redundancy and emphasise pertinent inputs. From the perspective of learning theory, employing more features progresses the prediction accuracy but the practical evidence suggests that this is not always correct since not all features are necessary to identify the data class label. Feature selection [34] aids in the reduction of less significant features or irrelevant features and enhances algorithmic performance. Botox Optimization Algorithm (BxOA) is provided by the RFS-HWP model to help choose the important features most effectively for weather prediction. BxOA is a new metaheuristic optimization algorithm enthused by the botox operation process. The goal of BxOA is to tackle optimization issues by using a human-based tactic. The BxOA is mathematically investigated and designed by Botox treatments, where defects are identified and addressed to enhance appearance. Additionally, it is more capable of striking a balance between exploitation and exploration. Each and every person requesting Botox treatments is regarded as a member of BxOA. BxOA mimics how a doctor injects Botox into certain face muscles to minimize wrinkles and boost beauty. The BxOA strategy, like Botox is selecting decision variables and offering a specific value to enhance a candidate's solution.

$$Z = \begin{bmatrix} \hat{Z}_1 \\ \vdots \\ \hat{Z}_j \\ \vdots \\ \hat{Z}_P \end{bmatrix}_{P\times n} = \begin{bmatrix} z_{1,1} \cdots z_{1,f} \cdots z_{1,p} \\ \vdots \\ z_{j,1} \cdots z_{j,f} \cdots z_{j,p} \\ \vdots \\ z_{Q,1} \cdots z_{Q,f} \cdots z_{Q,p} \end{bmatrix}_{Q\times p} \tag{3}$$

The population members of BxOA are used as the features in optimal feature selection. Each member contributes to the values of decision variables conferring to their location in the problem-solving space,

which is signified as a vector. The population matrix from this vector, which includes the decision variables, is given by the equation Eq 3.

During initialization, the following equation is engaged to randomly assign each BxOA member's position.

$$z_{j,f} = LB_f t_{j,f}.(UB_f - LB_f),\ j = 1,2,\ldots,Q,\quad f = 1,2,\ldots,p \tag{4}$$

where, UB and LB signify the upper and lower bounds of f^{th} decision variables, Z designates the population matrix of BxOA, $\hat{Z}_j$ defines the j^{th} member of BxOA (candidate solution), p specifies the number of decision variables, $t_{j,f}$ denote the random numbers from the interval $[0,1]$, and Q represents the total population members, respectively.

The fitness function is employed to rate each feature's excellence (potential solution). The objective of BxOA is to identify the subset of ideal features in a search region with minimum feature subset size and lowest classification error rate. In the proposed method, the fitness function engaged for the feature selection problem is given as follows:

$$H_l = (1-\psi)\times Sa + \psi \times \left(\frac{Ns}{Dimen}\right) \tag{5}$$

where, Ns describes the size of the chosen feature subset, Sa defines the classification accuracy, ψ exemplifies the weight parameter set to 0.01 and, $Dimen$ indicates the dimension. The best value attained for the fitness is considered by the optimal member of BxoA. According to the BxOA's design, the number of face muscles that need Botox injections reduces as the algorithm iterates. For that reason, the total chosen muscles (i.e., decision variables) for injecting Botox is calculated using the following expression:

$$Q_d = \left\lceil 1 + \frac{p}{v} \right\rceil \le p \tag{6}$$

where, Q_d describes the total muscles that need Botox injections and v represents the iteration counter value. The doctor assesses the patient's face and wrinkles to find the best places for Botox injections. Due to this fact, the variables to be injected for each member of population are selected based on the equation below.

$$Cdv_j = \{g_1, g_2, ..., g_k, ..., g_{Q_d}\},\ g_k \in \{1,2,3,...,p\} \tag{7}$$

where, g_j specifies the location of k^{th} decision variable chosen for injecting Botox and Cdv_j resembles the set of potential decision variables for the j^{th} member of population.

The following formula in BxOA, which is similar to the doctor's choice in deciding the quantity of drug for Botox injection based on patient needs and experience, is used to calculate the Botox injection amount for each member:

$$\hat{D}_j = \begin{cases} \hat{Z}_{Mean} - \hat{Z}_j, & v < \dfrac{V}{2} \\ \hat{Z}_{Best} - \hat{Z}_j, & else \end{cases} \tag{8}$$

where, $\hat{D}_j = (d_{j,1}, ..., d_{j,k}, ..., d_{j,p})$ exemplifies the considered amount for botox injection to the j^{th} member, V suggests the total number of iterations, $\hat{Z}_{Best}$ states the best member of population, and $\hat{Z}_{Mean}$ implies the mean population position (which means $\hat{Z}_{Mean} = \frac{1}{Q}\sum_{j=1}^{Q} \hat{Z}_j$).

After a facial muscle botox injection, wrinkles vanish and changes the appearance of face. Using the following equation, a new location is first computed for each BxOA member based on botox injection:

$$\hat{Z}_j^{New} : z_{j,g_k}^{New} = z_{j,g_k} + t_{j,g_k} . d_{j,g_k} \tag{9}$$

where, $\hat{Z}_j^{New}$ represents the new location of j^{th} member after injecting botox, d_{j,g_k} signifies the dimension of botox injection for j^{th} member (i.e. $\hat{D}_j$), z_{j,g_k}^{New} resembles its g_k^{th} dimension, and t_{j,g_k} indicates a random number with uniform distribution. If the fitness function's value maximizes, this new location exchanges the resultant member's preceding location, as seen in the below equation:

$$\hat{Z}_j = \begin{cases} \hat{Z}_j^{New}, & H_j^{New} < H_j \\ \hat{Z}_j, & else \end{cases} \tag{10}$$

where, H_j^{New} signifies the value of fitness function. The pseudocode of feature selection using BxOA

is provided in Algorithm 1.

Algorithm 1: Feature selection using BxOA
Start
Initialize the size of population Q and total number of iteration V
Set the variables, fitness function, and constraints
Generate the initial population matrix in random
Determine the fitness function
Compute the best candidate solution $\hat{Z}_{Best}$
For $v = 1$ to V
Update the number of decision variables to inject botox based on Equation (6)
For $j = 1$ to Q
Express the variables that are reflected for botox injection by Equation (7)
Evaluate the amount of botox injection by Equation (8)
For $j = 1$ to Q_d
Compute the new position of j^{th} BxOA member by Equation (9)
End
Evaluate the fitness function based on $\hat{Z}_j^{New}$
Update the j^{th} member of BxOA by Equation (10)
End
Save best obtained candidate solution
End
Output the optimal features
Stop

3.3. HDGT-HNN based Weather Prediction

In the framework of weather prediction, the proposed model has utilized Hybrid Deep Gated Tobler's Hiking Neural Network (HDGT-HNN) to forecast different weather conditions like pressure, humidity and temperature. The HDGT-HNN model utilizes the selected features as their input and outputs various weather parameters compared to the actual observed values. The integration of CNN, Gated

Recurrent Unit (GRU), Additive Attention Mechanism (AAM) and Hiking Optimization Algorithm (HiOA) makes up the HDGT-HNN model. The architecture of HDGT-HNN model is provided in Figure 2.

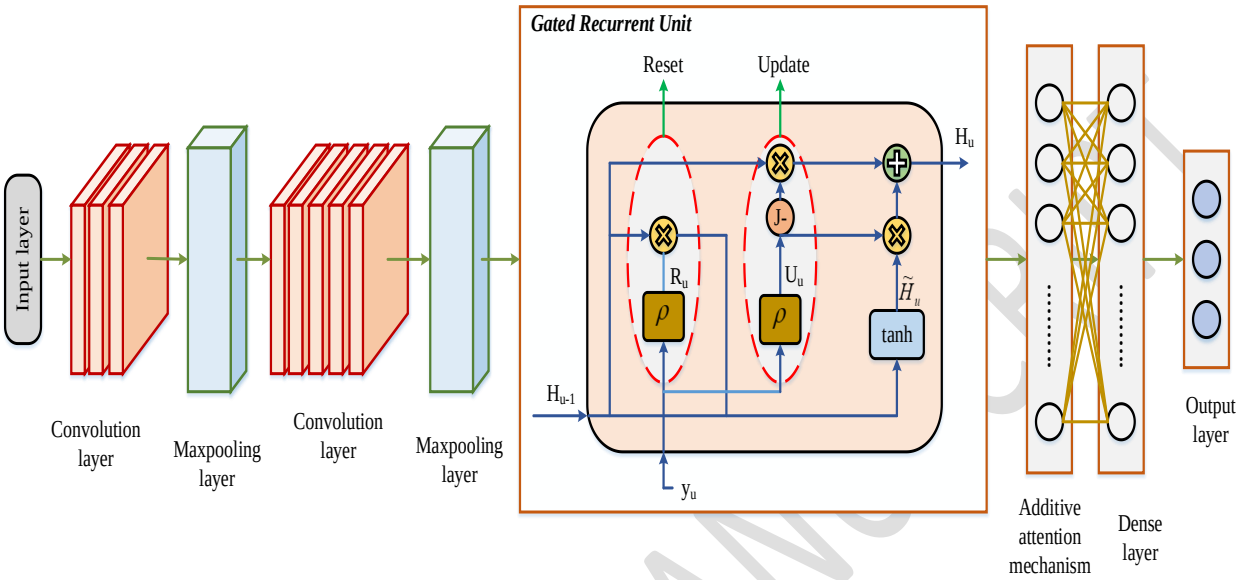


Figure 2: Architecture of HDGT-HNN model

First, the CNN is employed to record the features of weather in the input sequence. Then, this data is combined with an additive attention layer and a GRU to model the sequence. Convolutional kernels are used to process the input data and PReLU function has used to activate the data. The temporal information is effectively captured by utilizing the ability of GRU layer by integrating past, present, and future data. Moreover, the attention fusion is performed by combining the output sequences from the GRU model, the generated feature maps of CNN model and the output of additive attention layer. This is accomplished by applying an additive attention layer. Eventually, the combined features are subjected to a fully collected layer for linear transformation, model compilation, and output generation.

3.3.1. Convolutional Neural Network

One-dimensional CNNs are typically employed to handle 1D time series data or textual data since they are superior at extracting features from short, fixed-length inputs [35]. At first, the input sequence is scanned in a sliding fashion by means of a convolution kernel (or filter), followed by convolutional

operations to extract prominent features for making a new feature representation. The convolution operation output δ is calculated for the input sequence Y and the convolution kernel X:

$$\delta[k] = (Y.X)[k] = f\left(\sum_{l=0}^{L-1} Y[k \times U_t + l] X[l] + B_{cnn}\right) \tag{11}$$

where, $\delta[k]$ resembles the k^{th} element of output sequence ensuing from the convolution operation. The convolution kernel has been multiplied element-by-element with a part of input sequence (from k to $k+L+1$) and summed to create the k^{th} element of output sequence. In Equation (11), $Y[k+l]$ describes the $k+l^{th}$ element of input sequence, in which l symbolizes the counterweight at which the convolution kernel slides and k represents the current position of convolution operation. Besides, U_t indicates the time step, L states the convolution kernel's window size, B_{cnn} signifies a bias term, and $X[l]$ defines the l convolution kernel weight, which is utilized to execute element-by-element multiplication. Also, $f(.)$ characterizes the activation function that provides nonlinear properties. Next, the feature map size is minimized by a pooling procedure, which instantaneously extracts the significant features and reduces the model's computational intricacy. The following is the calculating formula:

$$y[k] = Maxi(Y[k \times U_t : (k+1) \times U_t]) \tag{12}$$

where, the feature sequence attained by the pooling process is characterized by first k elements, or $y[k]$. In Equation (12), the subsequence that has chosen in the input sequence is designated by $Y[k \times U_t : (k+1) \times U_t]$. This sequences start index is given as $k \times U_t$, and its end index is stated as $(k+1) \times U_t$. This type of indexing action is normally utilized for intercepting a certain part of the window or sequence.

3.3.2. Gated Recurrent Unit

An improved version of LSTM network is termed as GRU network. It combines hidden state and neuron state, optimizes the three LSTM gate architectures, and unifies input and forget gates into a single update gate. It can successfully diminish the gradient disappearance of RNN and lower the LSTM network unit's parameter. As a result, GRU falls the model's training time.

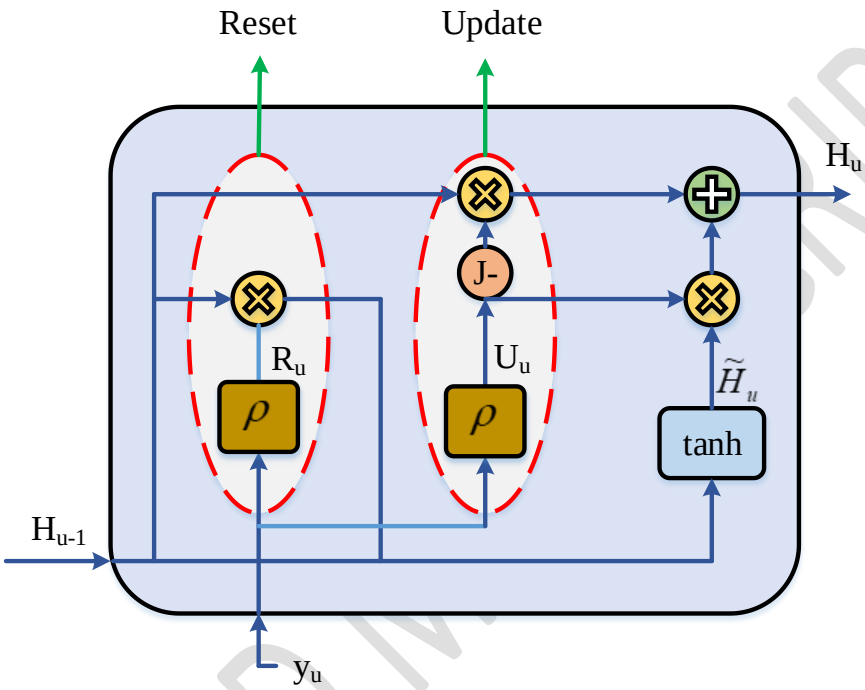


Figure 3: Structure of GRU

Figure 3 depicts the structure of GRU network. The mathematical formulas of GRU are explained below.

$$R_u = \rho(X_R.[H_{u-1}, y_u]) \tag{13}$$

$$U_u = \rho(X_U.[H_{u-1}, y_u]) \tag{14}$$

$$\tilde{H}_u = \varphi(X_{\tilde{H}}.[R_u \times H_{u-1}, y_u]) \tag{15}$$

$$H_u = (J - U_u) \times H_{u-1} + U_u \times \tilde{H}_u \tag{16}$$

$$z_u = \rho(X_o.H_u) \tag{17}$$

where, y_u represents the input vector, H_u defines the state memory variable at the current moment, R_u resembles the state of reset gate, H_{u-1} indicates the state memory variables at the previous moment, $\tilde{H}_u$ denotes the state of present candidate set, U_u signifies the state of update gate, and z_u implies the output vector of present moment. The weight matrices for the appropriate network activation function's input are resembled by X_R, X_U, $X_{\tilde{H}}$ and X_o. The Identity matrix is signified by J, matrix cross product is characterized by $\times$, vector connection is represented by $[]$, tanh activation function is denoted by φ, and the sigmoid activation function is denoted by ρ. The following is the mathematical definition of ρ and φ:

$$\rho(y) = \frac{1}{1+e^{-y}} \tag{18}$$

$$\varphi(y) = \frac{e^{y} - e^{-y}}{e^{y} + e^{-y}} \tag{19}$$

The update and reset gates are the primary components of GRU network. Following sigmoid nonlinear transformation, which governs the degree to which the state variable has brought into the present state at the previous time, the update gate receives the splicing matrix of the input variable y_u and the state memory variable H_{u-1} at that moment. The amount of data written to the candidate set previously is controlled by the reset gate.

3.3.3. Additive Attention Mechanism

AAM also known as multilayer perceptron attention is one of the attention mechanism that was initially used in sentence translation. It provides the model with the capability to assign different weights to various segments of input sequence. This allows for the sequence data to be processed with a focus on important elements. After a linear transformation, the input features to be summed and the transformed features to be activated by a sigmoid function for characterizing the similarity relationship between

the two features, which can handle with nonlinear relationships efficiently. The time series features are processed using CNN, and the data is encoded by utilizing GRU. Then, the output is taken as input and decoded by means of additive attention. The expression for additive attention is provided below as follows.

$$z_j = DecoderOuput(t_u, d_u) \tag{20}$$

where, z_j resembles the output of decoder at time step j, d_u states the weighted summation of encoder's encoding outcomes, and t_u describes the hidden vector of additive attention decoder at time step u. The decoder determines d_u at time step u based on the encoding outcome $(H_0, H_1, H_2, ..., H_u)$ of encoder as follows:

$$d_u = \sum_{j=1}^{U} b_{u,j} H_j \tag{21}$$

Concerning the weight, $b_{u,j}$ is computed as follows:

$$b_{u,j} = \frac{\exp(f_{u,j})}{\sum_{k=1}^{U} \exp(f_{u,k})} \tag{22}$$

where, $b_{u,j}$ defines the attentional weight that signifies the attentional weight of decoder from u time step to j time step, is derived from the attention score normalized through softmax. The expression of attention score is determined as follows:

$$f_{u,j} = \eta_b^U \tanh(X_b H_j + V_b t_{u-1}) \tag{23}$$

In additive attention, the decoder and encoder's hidden states are linearly transformed, and then a hyperbolic tangent function is employed to define the attention score. At the current position, the decoding outcome is to be subjected to the decoder's latent state at the previous time step t_{u-1}. The encoding outcome of the encoder at j^{th} time step is stated as H_j. $f_{u,j}$ defines the attention score

from the decoder at u^{th} time step to the encoder at j^{th} time step. X_b and V_b resemble the matrices of model's learnt parameter while η_b specifies the learned weight vector

3.3.4. Hyper-Parameter Tuning

Hyper-parameter tuning is considered as the process of determining the best values of hyper-parameter for a deep learning model. Building a high-performance model and selecting an appropriate model based on computational resources and time restrictions are essential. The design variables encompass the quantity of hidden layers, the number of neurones per layer, the learning rate, the dropout rate and the activation functions in the Hybrid Deep Gated Tobler's Hiking Neural Network (HDGT-HNN). HiOA is a new metaheuristic algorithm that draws inspiration from hiking, a well-liked recreational activity in the recognition of the similarities between the search landscapes of optimization issues and the rugged terrains traversed by hikers. Tobler's Hiking Function (THF), which calculates hikers' (or agents') walking velocity by considering the terrain's elevation and distance traveled is the basis of HiOA model. THF is used to find the positions of hikers when resolving an optimization issue. The HiOA facilitates a number of benefits when it comes to efficiency with computational resources and time constraints. Its design offers fast convergence to quickly converge, which makes it very useful in scenarios where time is significant. HiOA shortens the time needed to locate solutions by effectively navigating the search space compared to traditional optimization algorithm. Besides, it uses fewer resources and can handle large-dimensional issues without the significant computational overhead that comes with conventional algorithms. HiOA can be applied in a variety of fields without requiring extensive reconfiguration due to its adaptability to several optimization issues and thereby saving resources and time. Additionally, HiOA efficiently prevents from stuck in local optima, which enhances its ability to locate global optima faster with less computing effort. Owing to these

advantages, the proposed RFS-HWP model has employed HiOA to fine-tune the classification model's hyper-parameters. Hyper-parameter tuning uses the population members of HiOA as the hyper-parameters. Now, the mathematical model of HiOA based on THF for parameter tuning is discussed. THF is deliberated as an exponential function that computes the speed of hiker based on the slope or steepness of trail or terrain. The expression of THF is provided as:

$$\omega_{j,u} = 6e^{-3.5\left|T_{j,u}+0.05\right|} \tag{24}$$

where, $T_{j,u}$ represents the slope terrain or trail' and $\varpi_{j,u}$ signifies the velocity of hiker j (i.e., in km/h) at time or iteration u. Furthermore, the slope $T_{j,u}$ is provided by

$$T_{j,u} = \frac{di}{dy} = \tan\theta_{j,u} \tag{25}$$

Where, the variables dy and di signify the distance traveled by hiker' and the elevation difference, respectively. Besides, $\theta_{j,u}$ characterizes the terrain's or the trail's angle of inclination within range $[0,50^0]$.

HiOA takes the advantage of both hiker's social thinking as group and individual hiker's personal cognitive abilities. The lead hiker's location, the hiker's actual location, the sweep factor, and the initial velocity determined by THF contribute to the actual or updated velocity of a hiker. Thus, the present velocity of hiker j is provided by

$$\omega_{j,u} = \omega_{j,u-1} + \lambda_{j,u}(\alpha_{best} - \beta_{j,u}\alpha_{j,u}) \tag{26}$$

where, $\lambda_{j,u}$ signifies an integer with a uniform distribution in the interval $[0,1]$, $\varpi_{j,u}$ and $\varpi_{j,u-1}$ represent the initial velocity and current velocity of hiker j. $\beta_{j,u}$ signifies the sweeper factor in range $[1,3]$, and α_{best} indicates the lead hiker's location. In order to perceive where the lead hiker is headed and receive signals, sweep factor ensures the hiker doesn't wander too far from lead hiker.

Through the consideration of hiker velocity in Equation (24), updated position $\alpha_{j,u+1}$ of hiker j is obtained as follows:

$$\alpha_{j,u+1} = \alpha_{j,u} + X_{j,u} \tag{27}$$

The initial agent setup in a variety of metaheuristic algorithms, including HiOA is an important factor that has a great influence on how quickly convergence is reached and the feasible solutions can be found. Although there are other strategies, such as heuristic-based or problem-specific initialization approaches, the HiOA uses the random initialization methodology to initialize the locations of its agents. The below equation characterizes the lower bound φ_k^1 and upper bound φ_k^2 of the solutions, which determines the initialization of locations $\alpha_{j,u}$.

$$\alpha_{j,u} = \varphi_k^1 + \chi_j(\varphi_k^2 + \varphi_k^1) \tag{28}$$

Where, δ_k indicates the uniformly distribution integer in the interval $[0,1]$. The parameter sweep factor influences the HiOA's tendency for exploration and exploitation. As seen in Equation (26), this factor has a significant influence on the gap between the trail leader and other hikers. Furthermore, the HiOA's intrusive and exploitative behaviors are considerably shaped by the trail's slope, which impacts hiker velocity as exposed by Equation (24) and Equation (25). Besides, the HiOA tends to lean more toward an exploitation phase as the sweep factor range is extended. Whereas, lowering the SF range generally promotes an exploratory stage in HiOA. Additionally, the HiOA is steered toward the exploitation phase if the trail's degree of slope is reduced. All these factors collectively contribute to influence HiOA performance while tackling optimization issues.

4. Experimental Results and Discussion

This section presents the outcomes of RFS-HWP model and highlights its advantages over state-of-the-art methods for accurate weather prediction. The RFS-HWP model is implemented using Python platform to do experiments of weather prediction on a personal computer (PC) with 16 GB RAM and Intel(R) Core (TM) i5-4670CPU@3.20GHz processor, which operates on a 64-bit operating system. For experimenting the weather prediction process, the RFS-HWP model utilized a publicly available Jena climate dataset. The loss is computed using quadratic loss function and mean squared error (MSE). Optimizer employed in RFS-HWP model is HiOA. The learning rate is set to 0.01. To prevent

overfitting, the layer associated with dropout rate is utilized. Several other performance measures are analyzed, and proves the efficacy of RFS-HWP model over existing approaches. The following sections deliver the justifications of dataset employed, performance metrics, and result discussion.

4.1. Dataset Description

The RFS-HWP model has employed publically available time series Jena climate dataset for experimentation. The meteorological station located at the Max Planck Institute of Biogeochemistry in Jena, Germany, is the source of Jena climate dataset. It has a comprehensive assemblage of 14 features, each carefully recorded at every 10 minutes. This massive data collection was spanned over the course of eight years, commencing on January 1, 2009 and ending on December 31, 2016. There are 420,551 timestamps in this interval, and each one is related to 14 meteorological parameters. The dataset is available in https://www.kaggle.com/datasets/mnassrib/jena-climate. The features of Jena climate dataset is provided in Table 1.

Table 1: Features of Jena climate dataset

Features	Description	Format
Date time	Date-time reference	01.01.2009 00:10:00
T (degC)	Temperature in Celsius	-8.02
p (mbar)	Pascal SI derived a unit of pressure	996.52
Tdew (deg C)	Temperature in Celsius relative to humidity	-8.9
Tpot (K)	Temperature in Kelvin	265.4
VPmax (mbar)	Saturation vapor pressure	3.33

rh (%)	Relative humidity	93.3
VPdef (mbar)	Vapor pressure deficit	0.22
VPact (mbar)	Vapor pressure	3.11
H2OC (mmol/mol)	Water vapor concentration	3.12
sh (g/kg)	Specific humidity	1.94
wv (m/s)	Speed of wind	1.03
rho (g/m^3)	Airtight	1307.75
wd (deg)	Direction of wind in degrees	152.3
max. mv (m/s)	Wind speed at maximum	1.75

4.2. Performance Measures

Several performance measures, including accuracy, F1-score, MSE, recall and precision with their mathematical formulas are offered in this section. The ratio of accurate weather condition predictions to complete data elements is known as prediction accuracy. Recall is replicated as the ratio of true positive (TP) results to the total number of matters in the positive class. Precision, or positive predicted value (PPV), mentions to the percentage of optimistic weather predictions that belong to every positive class. The F1-score is a measure that thoroughly replicates the average of recall and PPV. The average squared difference between the actual and predicted values is designated MSE. The below expression

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{29}$$

$$\Re ecall = \frac{tp}{tp + fn} \tag{30}$$

$$Precision = \frac{tp}{tp + fp} \tag{31}$$

$$F1-score=\frac{2*Precision*\Re ecal}{Precision+\Re ecall} \quad (32)$$

$$MSE=\frac{1}{p}\sum_{j=1}^{p}(z_j-\hat{z}_j)^2 \quad (33)$$

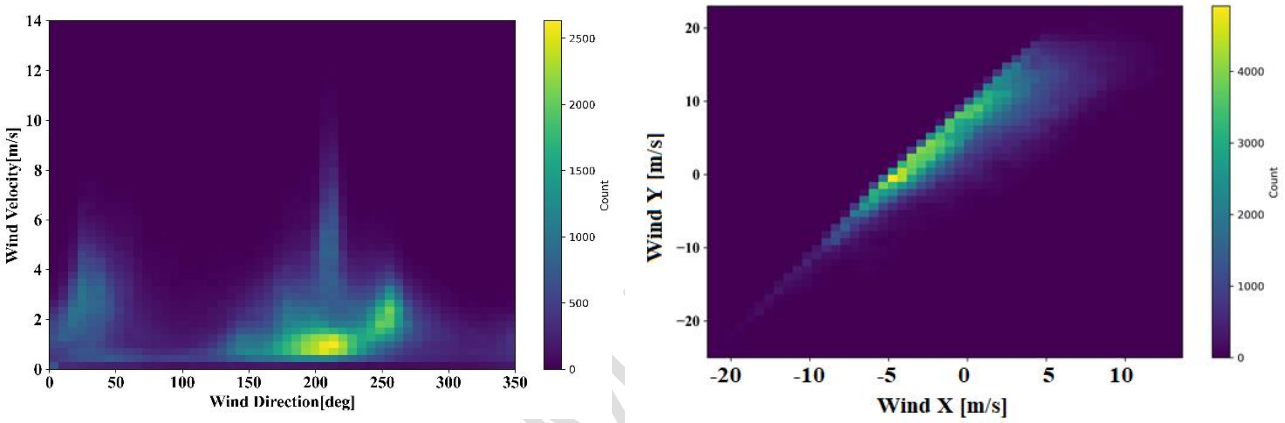
where, fp designates false positive, tn states true negative, tp indicates TP, and fn symbolizes false negative. Furthermore, p signifies the number of observation, z_j designates the actual value, and $\hat{z}_j$ resembles the predicted value.

4.3. Performance Evaluation

This section covers a detailed evaluation and result comparison for predicting weather conditions. The RFS-HWP model produces outputs which serve two purposes because it meets requirements for both continuous and discrete weather forecasting functions. Three meteorological elements, temperature, pressure and humidity serve as the basis for the model to predict weather conditions. The model employs numerical simulation for predicting meteorological variables which include temperature and pressure together with humidity data. The assessment metrics for this study include Mean Squared Error (MSE) to determine the amount of deviation between actual and predicted observations. The continuous output functions enable precise specification forecasting which proves necessary for exact climate measurement systems. RFS-HWP transforms meteorological condition variables into specific classifications through established criteria which derive from continuous data analysis. The model determines whether condition categorization by making forecasts about temperature levels as well as pressure variations and humidity patterns. The performance assessment of these categories relies on the Accuracy, Recall, Precision and F1-score indices for determining the model accuracy in predicting categorized weather results.

The RFS-HWP model considers three classes such as humidity, temperature and pressure. These classes resemble the key meteorological variables that the RFS-HWP intends to forecast correctly.

Figure 4 indicates the distribution of wind direction and wind vector over a specific period. As the wind can affect temperature distribution and humidity level, this aids to understand the overall dynamics and considerably influence the patterns of weather. Figure 4(a) describes the direction of wind in degrees and it displays the orientation from which wind is blowing. In Figure 4(b), the wind vector derived from wind direction data is shown. The plot wind vector visually resembles both the magnitude and direction of wind and permits of understanding the wind behavior.



(a) (b)

Figure 4: Distribution of wind direction and wind vector

The graphical representation has significantly captured the temporal variation in wind pattern, which is vital for predicting weather by plotting these parameters against time and date. This visualization supports in determining the trends in the behavior of wind and contributes into weather conditions and their influences on weather systems. Figure 5 displays the humidity distribution derived using Jena climate dataset. This graphical representation provides the variation in humidity over a specified time period and shows how the level of humidity varies throughout the day by emphasizing the patterns that can correlate with changes in atmospheric pressure and temperature. By plotting the value of humidity against time, the graphical representation exposes distinct trends, like peaks during certain hours, which can resemble greater moisture levels generally related with particular weather conditions. This

visualization is significant for understanding the humidity's temporal dynamics in Jena region and offers a valuable insight for forecasting and meteorological analysis.

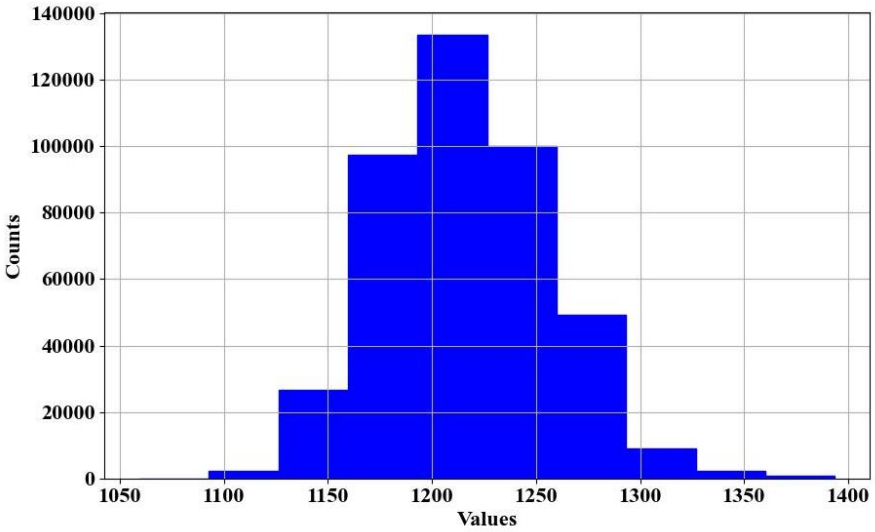


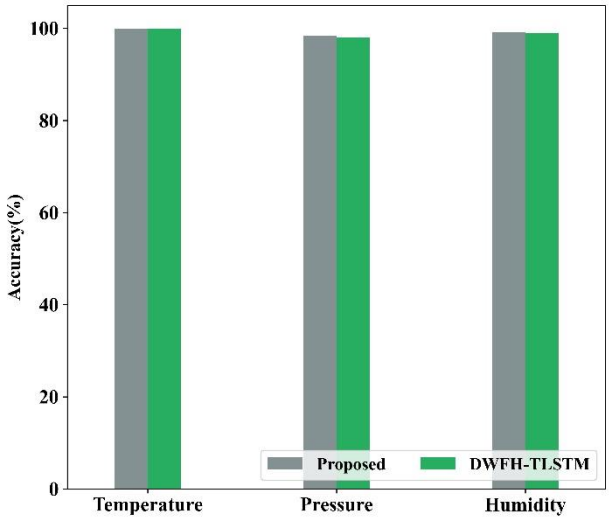
Figure 5: Distribution of humidity

Figure 6 provides a thorough analysis of different performance indicators for several weather classes predicted by the RFS-HWP model and existing DWFT-TLSTM [24] using Jena Climate dataset. Here, the three main meteorological variables such as pressure, temperature, and humidity are emphasized. A number of performance measures are given for each parameter, particularly precision, accuracy, recall, and F1 score, which jointly evaluate the predictive ability of RFS-HWP model. The accuracy metric shown in Figure 6(a) displays that the proportion of the model's predictions for each weather condition are accurate. Remarkably, the RFS-HWP model predicted temperature with an accuracy of 100%, which means that it accurately noticed every incident without creating any mistakes. This degree of exactness expresses how reliable the model is at predicting temperature, which is significant for many applications in climate science and meteorology.

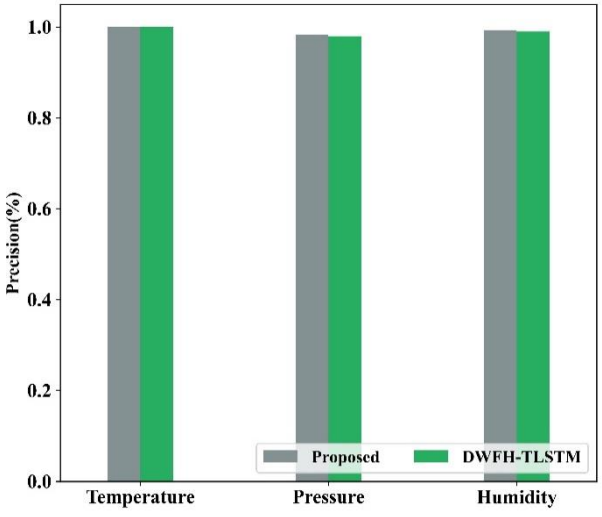
Comparative Analysis Between RFS-HWP and DWFT-TLSTM: DWFT-TLSTM (Dynamic Weighted Feature Transformation-Temporal Long Short-Term Memory) represents a deep learning technique that detects meteorological data spatial-temporal correlations. DWFT-TLSTM delivers outstanding

performance in humidity detection with 99% accuracy while reaching perfect agreement in other metrics and performing highly effectively for temperature forecasting. The system uses a dynamic weight control mechanism which establishes the most important factors as top priorities. DWFT-TLSTM demonstrates two weaknesses such as heightened computational requirements together with diminished ability to generalize pressure measurements with an F1-score of 97.8%.

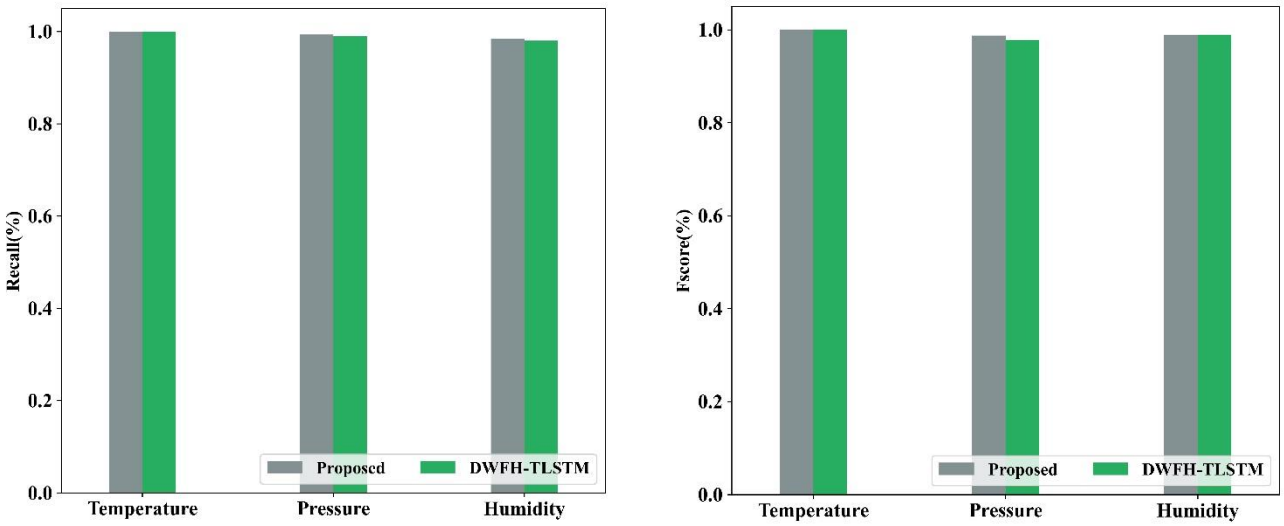
The model RFS-HWP demonstrates increased total performance through its 99.3% accuracy together with F1-scores of 98.79% for pressure measurement and 100% accuracy and 99.12% for temperature and humidity measurements. BxOA as a Botox Optimisation Algorithm together with HiOA as Hiking Optimisation Algorithm ensures both computer efficiency and reliable results through their powerful feature selection and effective hyperparameter tuning functions. The RFS-HWP system performs better across weather patterns yet DWFT-TLSTM excels at timing patterns detection.



(a)



(b)



(c) (d)

Figure 6: Comparison in terms of accuracy precision recall and f1-score

The RFS-HWP model demonstrated high accuracy rates of 99.21% and 98.45% for both humidity and pressure, demonstrating its efficacy in forecasting these parameters. Whereas, the exiting DWFT-TLSTM has also attained an accuracy of 100% for temperature, 99% for humidity and 98% for pressure because of capturing spatial and temporal relationships. On the other hand, the overall accuracy value attained by the proposed RFS-HWP model is 99.3%, whereas the existing DWFT-TLSTM attained 98.2%. Along with accuracy, the graphical representation encompasses precision, f1-score and recall metrics for establishing deeper insights into the model's performance. Precision clearly expressed the model's ability to prevent FPs by counting the percentage of actual positive predictions among all positive predictions. With a precision of 100% for temperature, the model accurately identified every instance of temperature, as shown in Figure 6(b). The precision values for pressure and humidity are 98.45% and 99.32%, and the existing method has achieved 98% and 99%. This describes that the model minimized FP predictions while maintaining a high accuracy level. Another significand metric recall, sometimes describes to as sensitivity, evaluates the RFS-HWP model's ability to determine all relevant instances. As perceived in Figure 6(c), the RFS-HWP model has able to

clearly resemble almost all real instances of these weather parameters with better recall values of 99.3% for pressure, 100% for temperature, and 98.5% for humidity. The exiting DWFT-TLSTM has achieved 99% for pressure, 100% for temperature, and 98% for humidity. The RFS-HWP model's excellent performance in all three weather conditions is further maintained by F1 score, which is depicted in Figure 6(d). A balanced capability to predict with no bias towards FPs or FNs is highlighted by the proposed model's f1-scores of 98.79% for pressure, 100% for temperature, and 99.12% for humidity. The exiting DWFT-TLSTM has achieved f1-score of 100% for temperature, 97.8% for pressure, and 98.9% for humidity. Conversely, the overall precision, recall and f1-score value attained by the proposed RFS-HWP model is 99.25%, 99.4% and 99.29%, whereas the existing DWFT-TLSTM has attained 98%, 99% and 97%.

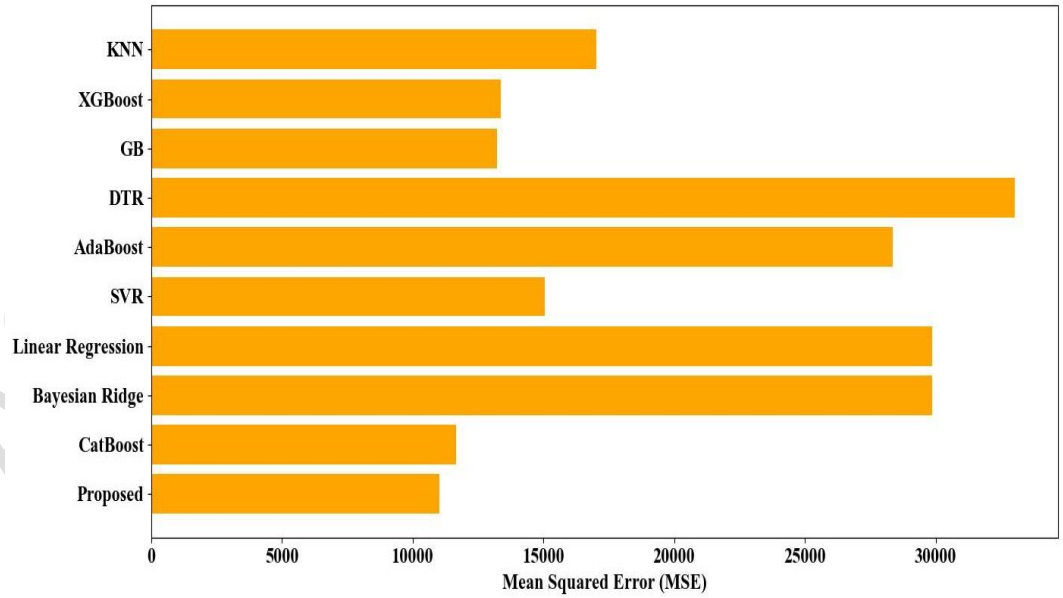


Figure 7: Comparison in terms of MSE

Figure 7 offers the comparison of various machine learning (ML) models in terms of MSE metrics for weather prediction. AdaBoost, Decision Tree Regression (DTR), Gradient Boosting (GB),

XGBoost, K-Nearest Neighbors (KNN), Linear Regression, Support Vector Regression (SVR), Bayesian Ridge, and CatBoost are the techniques used for comparison [24]. The proposed RFS-HWP model has attained minimum MSE score of 11,374.73. As compared to the existing method, CatBoost outperformed the other ML algorithms in weather prediction as evidenced by its minimum MSE of 11032. On the other hand, the DTR approach has the highest MSE of 33,014.21, resembling a lower level of accuracy in predicting weather. Thereby, the RFS-HWP model is effective in achieving reduced MSE values and greater prediction accuracy in weather forecasting tasks, as demonstrated by the graphical depiction that resembles the variability in performance across different ML approaches.

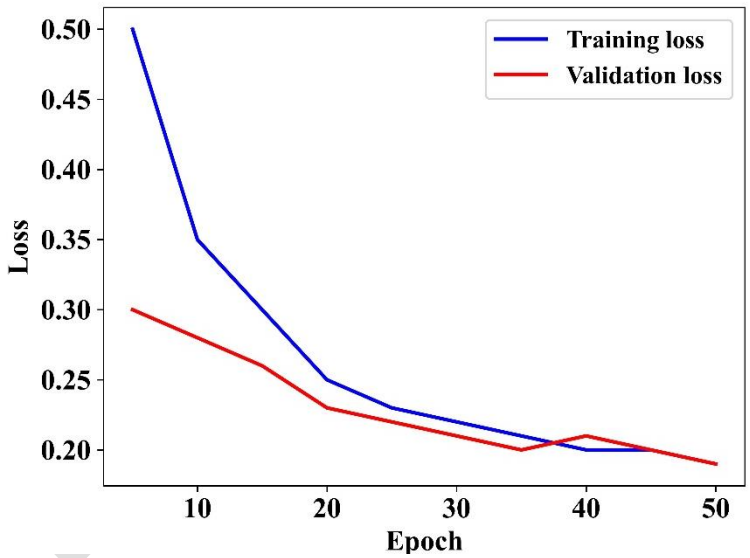
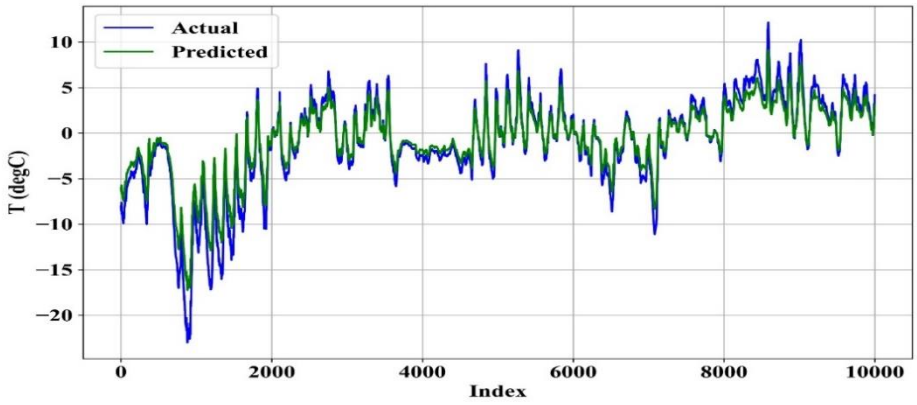
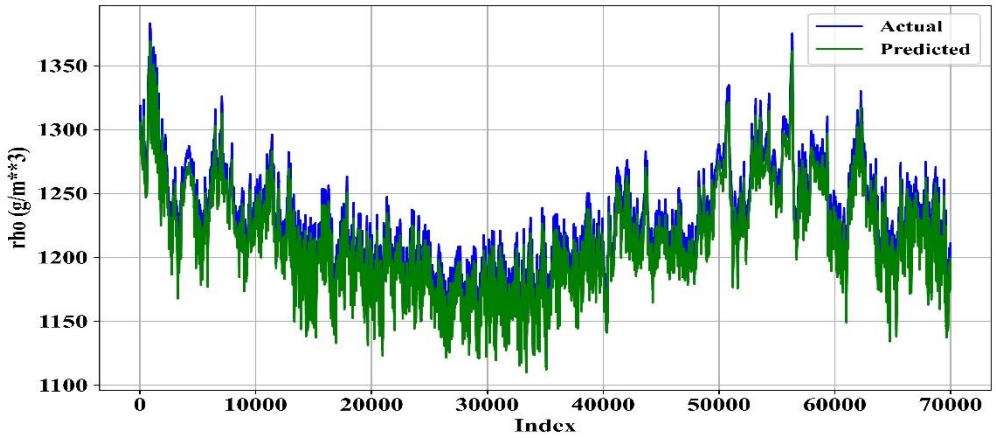


Figure 8: Analysis of training and validation loss

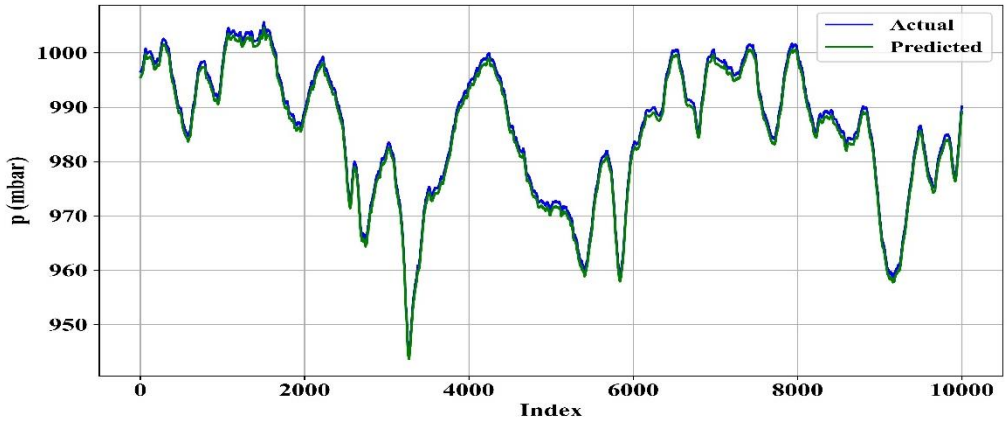
The training and validation loss analysis of RFS-HWP model is provided in Figure 8. This figure exemplifies the performance of predictive model over training epochs and depicts how the loss values varies as the model learn from the data. The error calculated on the training dataset is represented as training loss and the validation loss signifies the error on validation set. The figure shows that the predictive model is effectively learning and generalizing from data since there is a decreasing trend in both validation and training loss. Besides, the model exhibits reduced overfitting issues and offers insights into predictive abilities for weather prediction.



(a)



(b)



(c)

Figure 9: Analysis in terms of actual versus predicted values (a) temperature (b) humidity (c) pressure

Figure 9 presents the comparative analysis of actual and predictive temperature, humidity and pressure values. The figure 9(a) states the RFS-HWP's performance in predicting temperature by

plotting the actual recorded temperature alongside the predicted temperature values over a specified time period. The figure 9(b) and figure 9(c) state the RFS-HWP's performance in predicting humidity and pressure by plotting the actual recorded humidity and pressure alongside the predicted humidity and pressure values over a specified time period. These comparisons permit for an assessment of reliability and accuracy of the RFS-HWP model. The predictive model is effective when the actual values align closer to the predicted values. In the graphical representation, it is perceived that the predicted value of RFS-HWP is closer to the actual value. Also, the graphs highlighted discrepancies, representing periods where the model's predictions differ from actual observations, which supports to determine the weakness and strength of model.

Conclusion

This paper contributes to a unique RFS-HWP model for predicting different weather conditions. To ensure data quality, the RFS-HWP model begins by pre-processing input dataset with Z-score normalization and imputation for missing data. To improve the precision rate with minimum complexity, the best subset of features is selected by means of BxOA. The RFS-HWP classifies weather conditions into categories of temperature, pressure and humidity. In order to maximize the model performance, the HiOA is also utilized to fine-tune the hyper-parameters of regression model, HDGT-HNN. Thus, the comprehensive method that incorporates feature selection, pre-processing, and prediction demonstrates that the RFS-HWP can effectively predict the weather conditions. This helps to progress public safety, agricultural planning, transportation and environmental protection. The performance of RFS-HWP model is evaluated by means of different performance indicators, and it accomplishes 99.3% classification accuracy, 99.25% precision, 99.4% recall and 99.29% f1-score. The proposed RFS-HWP model outperforms state-of-the-art techniques in all performance measures. Even

with improved performance, the performance of model could vary across different climatic conditions or geographical regions and minimizes the accuracy. Future work involves adding real-time data to predictive model for improving the model's responsiveness utility in practical applications. Furthermore, experimenting with other optimization algorithms for feature selection and hyper-parameter tuning can disclose even more efficient neural network configurations. Lastly, the model's strength and generalizability will be verified by validating it on a number of datasets from different geographic regions with additional meteorological variables.

Ethics Declarations.

Funding statement:

No funding was received for this study.

Data Availability Statement

The datasets generated and or analyzed during the current study is publicly available of the submitted research work.

https://www.kaggle.com/datasets/mnassrib/jena-climate.

Conflict of Interest

The authors declare they have no conflicts of interest to report regarding the present study.

References

1. Chen, L., Bocheng, H., Xuesong, Wang., Jiazhen, Zhao., Wenke, Yang., and Zhengyi, Yang., (2023). Machine learning methods in weather and climate applications: A survey, Applied Sciences, 13, 12019.

2. Jaseena, K. U., and Binsu C. Kovoor., (2022). Deterministic weather forecasting models based on intelligent predictors: A survey. Journal of king saud university-computer and information sciences 34, 3393-3412.

3. Nirmal Kumar, M., Subramanian, P. and Surendran, R. (2025). Multivariate time series weather forecasting model using integrated secondary decomposition and Self-Attentive spatio-temporal learning network, Global NEST Journal.

4. Benavides, C., Llinet, Rodrigo, A. E. S., Miguel Ángel, M. C., and Calimanut-I. C., (2022). Review on spatio-temporal solar forecasting methods driven by in situ measurements or their combination with satellite and numerical weather prediction (NWP) estimates, Energies 15(12), 4341.

5. Brotzge, Jerald A., Don Berchoff, DaNa L. Carlis, Frederick H. Carr, Rachel Hogan Carr, Jordan J. Gerth, Brian D. Gross et al., (2023). Challenges and opportunities in numerical weather prediction. Bulletin of the American Meteorological Society, 104, E698-E705.

6. Soldatenko, S. A. (2024). Artificial Intelligence and Its Application in Numerical Weather Prediction, Russian Meteorology and Hydrology, 49, 283-298.

7. 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).

8. Guo, H., Yongjie, M., Zufeng, L., Qingzhi, Z., and Yuan Z., (2024). The Evaluation of Rainfall Forecasting in a Global Navigation Satellite System-Assisted Numerical Weather Prediction Model, Atmosphere 15(8), 992.

9. Hewage, P., Marcello T., Ella, P., and Ardhendu, B., (2021). Deep learning-based effective fine-grained weather forecasting model, Pattern Analysis and Applications, 24, 343-366.

10. Schultz, Martin G., Clara, B., Bing, G., Felix, K., Michael, L., Lukas Hubert, L., Amirpasha, M., and Scarlet, S., (2021). Can deep learning beat numerical weather prediction, Philosophical Transactions of the Royal Society A, 379, 20200097.

11. 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.

12. Bochenek, B., and Zbigniew, U., (2022). Machine learning in weather prediction and climate analyses—applications and perspectives, Atmosphere 13, 180.

13. Wu, Z., Gan, L., Zhile, Y., Yuanjun, G., Kang, L., and Yusheng, Xue., (2022). A comprehensive review on deep learning approaches in wind forecasting applications, CAAI Transactions on Intelligence Technology, 7, 129-143.

14. Hewage, P., Marcello, T., Ella, Pe., and Ardhendu, B., (2021). Deep learning-based effective fine-grained weather forecasting model, Pattern Analysis and Applications 24, 343-366.

15. Ajina, A., Christiyan, KGJ., Dheerej, N., and Kanishk, S., (2023). Prediction of weather forecasting using artificial neural networks, Journal of applied research and technology, 21, 205-211.

16. Ren, Xiaoli., Xiaoyong, Li., Kaijun, Ren., Junqiang, Song., Zichen, Xu., Kefeng, Deng., and Xiang, Wang., (2021). Deep learning-based weather prediction: a survey, Big Data Research, 23, 100178.

17. 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.

18. Dudukcu, H.V., Murat, T., Zehra Gulru, C. T., and Tulay, Y., (2023). Temporal Convolutional Networks with RNN approach for chaotic time series prediction, Applied soft computing, 133 109945.

19. Jeong, S., Inyoung, P., Hyun Soo, K., Chul Han, S., and Hong Kook, K., (2021). Temperature prediction based on bidirectional long short-term memory and convolutional neural network combining observed and numerical forecast data, Sensors, 21, 941.

20. Suleman, Masooma, A. R., and Shridevi, S., (2022). Short-term weather forecasting using spatial feature attention based LSTM model, IEEE Access, 10, 82456-82468.

21. Avila Saccol, G., Bui, V. H., & Su, W. (2024). A Deep-Neural-Network-Based Surrogate Model for DC/AC Converter Topology Selection Using Multi-Domain Simulations. Energies, 17(24), 6467,

22. Ouali, R., Legry, M., Dieulot, J. Y., Yim, P., Guillaud, X., & Colas, F. (2024). Convolutional Neural Network for the Classification of the Control Mode of Grid-Connected Power Converters. Energies, 17(24), 6458.

23. Bento, M. E. C. (2024). Physics-Informed Neural Network for Load Margin Assessment of Power Systems with Optimal Phasor Measurement Unit Placement. Electricity, 5(4), 785-803.

24. Mohammad, A. T., & Al-Shohani, W. A. (2024). Short-Term Prediction of the Solar Photovoltaic Power Output Using Nonlinear Autoregressive Exogenous Inputs and Artificial Neural Network Techniques Under Different Weather Conditions. Energies, 17(23), 6153.

25. Bento, M. E. C. (2024). Load Margin Assessment of Power Systems Using Physics-Informed Neural Network with Optimized Parameters. Energies, 17(7), 1562.

26. Utku, A., and U. Can., (2023). An efficient hybrid weather prediction model based on deep learning International Journal of Environmental Science and Technology, 20, no. 10, 11107-11120.

27. Teixeira, Rita., Adelaide, C., Eduardo, J., Solteiro, P., and Jose, Baptista., (2024). Enhancing Weather Forecasting Integrating LSTM and GA, Applied Sciences, 14, 5769.

28. Medina, S., Alfonso, A., and Leopoldo Eduardo C.B., (2022). Optimal design of hybrid renewable energy systems considering weather forecasting using recurrent neural networks, Energies, 15, 9045.

29. Venkatachalam, K., Pavel Trojovský, Dragan Pamucar, Nebojsa Bacanin, and Vladimir Simic. "DWFH: An improved data-driven deep weather forecasting hybrid model using Transductive Long Short Term Memory (T-LSTM)." Expert systems with applications 213 (2023): 119270.

30. Yalçın, S., (2022). Weather parameters forecasting with time series using deep hybrid neural networks, Concurrency and Computation: Practice and Experience, 34, e7141.

31. Liu, Y., Tharam, D., Wenjin, Yu., Wenny, R., and Fahed, M., (2020). Missing value imputation for industrial IoT sensor data with large gaps, IEEE Internet of Things Journal, 7, 6855-6867.

32. Arcía, Salvador., Julián, Luengo., and Francisco, Herrera., (2015). Data preprocessing in data mining. Springer International Publishing, 72, 17-33.

33. Muthulakshmi, P., and Parveen, M., (2023). Z-Score Normalized Feature Selection and Iterative African Buffalo Optimization for Effective Heart Disease Prediction, International Journal of Intelligent Engineering & Systems 16, 11-27.

34. Subbiah, S. S., Senthil Kumar, P., Karmel, A., Saminathan, S., and Muthamilselvan, T., (2023). Deep Learning for Wind Speed Forecasting Using Bi-LSTM with Selected Features, Intelligent Automation & Soft Computing, 35, 1-27.

35. Kiranyaz, S., Onur, A., Osama, A., Turker, Ince., Moncef, G., and Daniel, J. I., (2021). 1D convolutional neural networks and applications: A survey, Mechanical systems and signal processing, 151, 107398.