Int. J. Adv. Multidiscip. Res. (2023). 10(8): 20-38

International Journal of Advanced Multidisciplinary Research
ISSN: 2393-8870
www.ijarm.com
(A Peer Reviewed, Referred, Indexed and Open Access Journal)
DOI: 10.22192/ijamr Volume 10, Issue 8 -2023

Research Article

DOI: http://dx.doi.org/10.22192/ijamr.2023.10.08.003

Enhanced Weather Forecasting using the MeteroNet Model: A Comprehensive Ensemble Approach

Faiza Fareed Babar
Department of Artificial Intelligence
Hebei University of Technology
Tianjin, China
E-mail: *faizababar2016@yahoo.com*

Shi Lukui
Department of Artificial Intelligence
Hebei University of Technology
Tianjin, China
E-mail: *Shilukui@scse.hebut.edu.con*

Farah Farid Babar
Department of Computer and Engineering
University of Huddersfiled
Huddersfiled, UK
E-mail: *Farahbabar003@yahoo.com*

Fayaz Muhammad
Department of Computer Science
Sarhad University of Science and Information Technology
Peshawar, Pakistan
E-mail: *fayaz.khanr@gmail.com*

Abstract

Accurate weather forecasting is crucial for various domains, such as agriculture, transportation, and disaster management. However, due to the complex and dynamic nature of weather systems, achieving precise predictions remains challenging. This paper proposes a framework based on the MeteroNet model to enhance the accuracy of weather forecasting. The framework involves preprocessing weather data by numerically encoding non-numerical attributes and reducing data redundancy. A CNN network extracts features from the meteorological attributes, followed by feature selection using a genetic algorithm. The MeteroNet model overcomes limitations of individual models by combining LSTM, BiLSTM, and GRU through an ensemble approach. By ensembling multiple models, the MeteroNet model improves accuracy even with smaller training sets and combines models with different biases and error patterns for enhanced overall accuracy. The model employs a voting scheme using averaging Bayesian to determine the final prediction, assigning weights based on performance and reliability. Experimental results on a weather dataset from Peshawar, Pakistan demonstrate the superiority of the MeteroNet model over baseline models. The MeteroNet model achieved an accuracy of 96.22% and exhibited high precision, recall, F1-score from 87.50% to 100%, 89.80% to 100%, and 92.80% to 100% respectively across various weather classes. This research contributes to improving weather forecasting accuracy, benefiting numerous applications in diverse domains.

Keywords

Weather forecasting,
MeteroNet model,
Accuracy,
Ensemble approach,
Preprocessing

I. INTRODUCTION

In the current era of rapidly advancing technology, accurate weather forecasting has become a critical factor for various industries and sectors worldwide. The ability to predict weather conditions with precision holds immense value for sectors such as agriculture, transportation, energy, tourism, and disaster management. Reliable weather forecasts empower decision-makers to plan effectively, mitigate risks, optimize operations, and ensure public safety. However, weather prediction remains a challenging task due to the intricate and dynamic nature of atmospheric systems. In recent years, advancements in machine learning and ensemble techniques have shown promising results in improving the accuracy and reliability of weather forecasting models. Janizadeh et al. compared machine learning models for flash flood susceptibility mapping in the Tafresh watershed, with Random Forest achieving the highest accuracy. Key variables like elevation, slope, land use, and rainfall were identified as significant factors. However, the findings may have limited generalizability and did not consider socio-economic and human interventions' impact on susceptibility [1]. Mouatadid et al. focused on improving SPEI prediction using optimized data-driven models, highlighting limitations in generalizability, input data quality, and absence of external factors [2]. Zhang et al. presented DeepARIMA-LSTM for temperature prediction, outperforming traditional models but with limited generalizability and comparison to other deep learning models [3]. Hossain et al. achieved 97.97% accuracy in temperature prediction for Nevada using a deep neural network with stacked denoising auto-encoders, surpassing traditional neural networks. However, the study's focus is limited to Nevada, necessitating further research to enhance practical implementation and generalize the approach to other regions [4]. Salman et al. propose an LSTM-based weather forecasting model with intermediate variables, achieving a validation accuracy of 0.8060 and RMSE of 0.0775. However, the study lacks comparison with other models and utilizes only one dataset from Hang Nadim Indonesia airport [5]. Zaytar et al. utilized a multi-stacked LSTM deep learning approach to forecast weather parameters, including temperature, humidity, and wind speed. The study demonstrates the model's effectiveness and accuracy, indicating its potential for predicting additional weather parameters [6]. Pham et al. (Year) achieved 95.53% accuracy in forecasting daily maximum temperature using machine learning (ANNs, RF, SVM) on a single dataset, limiting generalization. Impact of additional variables and feature selection algorithms was not explored [7].

Karevan et al. proposed a hybrid model for temperature prediction, achieving high accuracy but limited to a single weather station in India [8]. Liang Ge et al. (2020) proposed advanced deep learning techniques for temperature forecasting, achieving high accuracy with an RSME of 0.17 and 96% accuracy in testing and 97% in validation. However, limitations included not considering factors like ventilation humidity and solar radiation [9]. Cifuentes et al. found that deep learning strategies had lower errors compared to traditional methods but highlighted the dependence on input combination, architecture, and learning algorithms [10]. Chattopadhysay et al. explored simpler techniques such as convolutional neural networks and logistic regression, which were less affected by reduced training set size [11]. Hrachya Astsatryan1 developed an advanced deep learning model for temperature forecasting using satellite data, achieving accuracy rates of 87.31% and 75.57% for short-term forecasts [12]. Suman Ravuri presented a deep generative model for probabilistic now-casting of precipitation, showing improved forecast quality and value, but with lower accuracy [13]. Bauer et al. developed a system for predicting extreme weather events using machine learning, acknowledging potential data limitations and the inability to account for unforeseen events or climate pattern changes [14]. Schultz et al. highlighted the benefits and challenges of machine learning in weather forecasting, stressing the importance of model validation and data quality for accurate forecasts [15]. Zhou et al. proposed a hybrid machine learning model for precipitation prediction, outperforming individual models with an accuracy of 91.79% [16]. Cristiano et al. highlight the challenges of rainfall forecasting using physical-based models due to the complexity of atmospheric processes and limited data availability, limiting their feasibility and accuracy [17]. Yalcin et al. propose a deep hybrid neural network approach for weather parameter forecasting, combining CNNs and LSTM networks to achieve superior prediction accuracy compared to traditional statistical models and other deep learning approaches. The inclusion of additional input features enhances accuracy, but limitations include the need for further research on hyperparameters and computational complexity [18]. Huu Nam Nguyen develops an AI model based on artificial neural networks (ANN) for rainfall prediction. However, there is room for improvement as indicated by the RMSE and MAE values, which are 0.8063 and 0.2487 for daily rainfall, and 0.8012 and 0.0731 for monthly rainfall [19]. Juliana Aparecida Anochi explores various machine learning models for precipitation prediction, but other models not mentioned can also be considered [20]. Scher et al. propose a computationally efficient machine learning approach using convolutional neural networks to predict weather forecast uncertainty. While the method outperforms alternative approaches, it has lower skill compared to ensemble weather forecast models in predicting uncertainty. The limited availability of past forecasts for training is a key limitation of the proposed method [21]. Verma et al. introduce a novel Stack and Bidirectional LSTM model with an intermediate variable for weather prediction. The model is evaluated using real-world weather datasets from India, demonstrating superior accuracy compared to other deep learning and statistical models. However, the study lacks discussion on implementation challenges and limitations of the proposed model in practical applications [22]. Yonekura et al. developed a deep neural network for short-term local rain and temperature forecasting. The study concluded that deep neural networks exhibited superior accuracy compared to other machine learning methods specifically for rain prediction [23]. Weyn et al. introduce a method for enhancing short-term temperature forecasting accuracy by integrating weather station and satellite remote sensing data. The combined data improved temperature forecasts, particularly during extreme temperature events. Although promising, the approach's computational intensity and data requirements may limit its feasibility for all organizations or regions [24].

Several studies have focused on ensemble techniques to enhance weather forecasting. Cho et al. proposed GRU (Gated Recurrent Unit) models, offer similar capabilities with reduced computational complexity. Data integration is another critical aspect in weather forecasting. Incorporating diverse data sources such as weather station data, satellite data, and remote sensing data has been shown to improve prediction accuracy [25]. In summary, the literature review demonstrates the importance of ensemble techniques, diverse model architectures, and data integration in weather forecasting. The proposed MeteroNet framework aligns with previous studies that have shown the benefits of combining models, such as LSTM, BiLSTM, and GRU, to enhance prediction accuracy. Additionally, the existing challenges in weather forecasting underscore the need for novel approaches that can overcome the limitations of individual models, limited generalizability, accuracy of predictions, and computational complexity. This research article aims to present the MeteroNet model, proposes an innovative ensemble framework that combines multiple models, leveraging their unique strengths and compensating for their weaknesses, overcoming accuracy reduction, addressing data generalizability limitations, and improving prediction reliability. The proposed approach represents a significant step forward in the field of weather forecasting, with the potential to impact various industries and decision-making processes. The paper is structured as follows. Section 2 presents the methodology, including the pre-processing of the acquired dataset and the division into testing and training. Section 3 discusses the results obtained from applying the classification algorithm. Finally, in Section 4, the paper concludes with a summary of the findings.

II. CONTRIBUTION TO THE WORK

The research makes significant contributions to the field of weather forecasting. It introduces the innovative MeteroNet framework:

- Overcomes limitations and enhances prediction accuracy. By combining LSTM, BiLSTM, and GRU models through an ensemble approach,
- The framework tackles accuracy reduction with smaller training sets. It also addresses generalizability and data limitations by integrating diverse models.
- Furthermore, the framework mitigates computational intensity and data requirements through distributed computational load. The introduction of a voting scheme using averaging Bayesian improves predictions by considering uncertainties and biases.

Overall, this research advances the state-of-the-art in weather forecasting and offers a comprehensive approach for accurate and reliable predictions.

III. METHODOLOGY

Weather forecasting using recurrent neural networks (RNNs) is a valuable approach with applications in diverse sectors. This methodology involves data collection, preprocessing, feature selection, and model training using LSTM, BiLSTM, and GRU models. Ensemble learning is utilized to enhance prediction accuracy. The performance is evaluated using classification metrics such as accuracy, precision, recall, and F1-measure. This comprehensive approach aims to improve weather prediction accuracy and support informed decision-making across various industries. Fig. 1. depicts the flow chart of the proposed methodology.

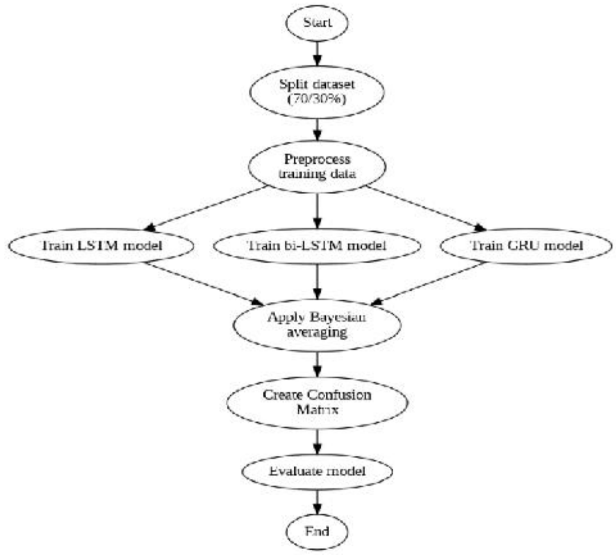

Fig. 1. Proposed Methodology Flow chart

A. *Data set collection and processing*

Data set preparation and preprocessing are essential for accurate weather forecasting models, involving data collection, cleaning, and transformation to enhance analysis and improve decision-making.

B. *Dataset*

The dataset used for weather forecasting encompasses various variables, including temperature, humidity, pressure, wind speed and direction, and precipitation. These variables are collected at regular intervals over a span of several years, and the quality and quantity of the data greatly influence the accuracy and reliability of the resulting forecasting models.

C. *Customize Dataset*

For this study, the dataset is tailored to Peshawar, a city in Pakistan's Khyber Pakhtunkhwa province, situated near the Afghanistan border. Peshawar experiences a subtropical climate characterized by hot summers and mild winters. The average temperature ranges from 34°C in summer (June to August) to 14°C in winter (December to February). The city receives the majority of its rainfall during the monsoon season (July to September), with an average annual precipitation of around 500mm. The dataset includes historical weather parameters such as temperature, humidity, atmospheric pressure, wind speed and direction, precipitation, cloud cover, solar radiation, air quality index, dew point, visibility, heat index, wind chill, thunderstorm probability, UV index, frost or freeze probability, storm surge potential, air mass, fronts, jet stream, convective outlook, tornado potential index, severe weather outlook, fire danger rating, ocean temperature, wave height, surf conditions, and rip current risk for the years 2018 to 2022. Some of the attributes from our custom data set as shown in table I.

TABLE I. RANDOM ATTRIBUTES OF CUSTOM CREATED WEATHER DATASET.

	Sample1	Sample2	Sample3
Temperature (C)	34.5	39.8	31.2
Humidity (%)	25	18	30
Atmospheric Pressure (hPa)	1012.2	1010.5	1011.8
Wind Speed (Km/h)	12	10	8
Wind Direction	NW	E	SW
Precipitation (mm)	0	0	0
Cloud Cover (%)	15	10	20
Solar Radiation (W/m2)	600	650	550
Air Quality Index	80	78	82
Visibility (Km)	10	11.5	9.2
Heat Index (°C)	37.2	42	35.8
Wind Chill (°C)	31.3	36.4	29.7
Air Mass	Tropical	Polar	Maritime

D. Data Preprocessing: Cleaning and Transforming Raw Data

Data preprocessing is a vital step in machine learning projects, involving cleaning and preparing raw data before model training. For weather forecasting with RNN, it includes tasks like removing missing data, normalizing features, and encoding categorical variables. Ordinal encoding is utilized to convert categorical variables (e.g., weather condition, wind direction) into numerical values. This ensures compatibility with machine learning algorithms. The resulting encoded data is then fed into the input layer of the CNN for feature extraction.

1) Ordinal Encoding: Converting Categorical Variables to Numerical Values

During data preprocessing, categorical variables like weather condition and wind direction are transformed into numerical values using ordinal encoding. This conversion enables machine learning algorithms to utilize these variables for decision-making. The output of the ordinal encoding phase serves as input to the CNN's input layer for feature extraction. Each feature, such as temperature, humidity, precipitation, etc., is assigned to specific nodes (M1 to M25) in the input layer.

2) Extracting Features with 1D Convolutional Filters

To extract features from the weather dataset, a CNN model is employed, utilizing 1D convolutional filters. These filters scan the numerical data, capturing local patterns at different scales. By sliding over the sequential data, the CNN can recognize important temporal patterns relevant to weather forecasting. The resulting feature maps contain higher-level representations that encode critical characteristics for accurate predictions. These feature maps undergo further processing through pooling and fully connected layers to aggregate and analyze the extracted features. The hierarchical transformation process and learned parameters optimize the CNN's ability to detect meaningful patterns in the input data.

The feature extraction process can be summarized with the following steps:

Convolution Operation:

$$c[i] = \Sigma[(j = 0) to (M - 1), X[i + j] * W[j]] \quad (1)$$

Activation Function:

$$c[i] = f(c[i]) \tag{2}$$

Pooling Operation:

$$P[i] = \max\left(C[iS: iS + K]\right) \tag{3}$$

Fully Connected Layer:

$$O = Wf.F \tag{4}$$

These operations enable the CNN to capture relevant features from the numerical weather data, considering the temporal nature of the dataset. The extracted features provide valuable information for subsequent layers to make informed predictions or classifications based on the sequential nature of the data.

3) Feature Selection with Genetic Algorithm

In feature selection for weather forecasting, we employ the Genetic Algorithm (GA) to identify the most relevant attributes from the CNN's output. The GA uses evolutionary principles, mimicking natural genetics, to optimize feature subsets. By evaluating fitness based on performance metrics and applying genetic operators like crossover and mutation, the GA iteratively generates candidate solutions. The fittest chromosome, representing the optimal feature subset, is selected as the final solution for improved weather prediction accuracy, shown in fig. 2.

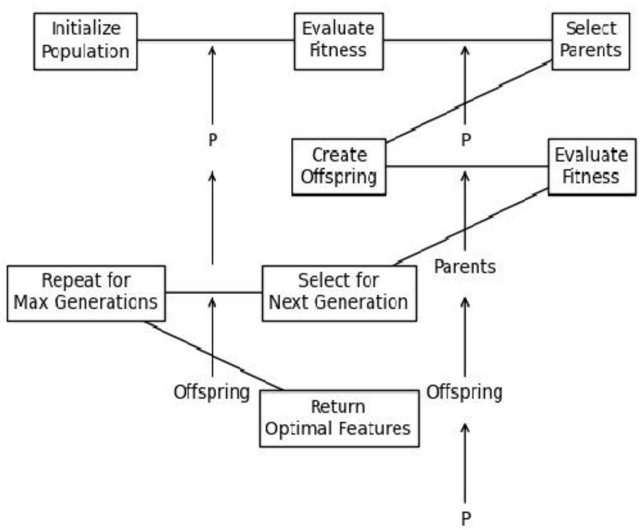

Fig. 2. A Visual Representation of the Genetic Algorithm Approach to Feature Selection.

The training dataset for weather prediction in Peshawar, Pakistan, consists of 20 observations taken at different times throughout the year. The dataset includes measurements of relative humidity, temperature, atmospheric pressure, wind speed, cloud cover, solar radiation, and air quality index. This sample is part of a larger dataset containing data from 20065 observations used for training and analysis. A data set sample shown in Table II.

TABLE II. SAMPLE DATA WEATHER PARAMETERS IN THE SAMPLE DATASET FOR PESHAWAR PARAMETER VALUES FOR TRAINING THE MODEL.

Date/Time	Humidity (%)	Average Temperature (C)	Atmospheric Pressure (mb)	Wind Speed (knot)	Cloud Cover (%)	Solar Radiation (W/m²)	Air Quality Index
1/1/2018 12:00	65	24.1	1011	1	20	150	86
3/15/2018 9:00	40	35	1010	2	30	200	75
6/22/2018 15:00	50	26	1008	1.5	40	300	92
9/12/2018 18:00	80	11	1012	1	60	250	120
12/1/2018 11:00	30	19	1015	1.5	10	100	65
2/23/2019 14:00	45	35	1011	2	20	150	80
5/10/2019 16:00	55	35	1008	1	50	350	95
8/3/2019 19:00	75	10	1009	2.5	70	300	130
11/16/2019 8:00	35	19	1013	1	15	120	70
1/19/2020 10:00	60	34	1010	1.5	25	150	85
4/2/2020 14:00	45	39	1009	2	35	250	90
7/5/2020 17:00	70	21	1007	1.5	60	400	120
10/8/2020 20:00	35	17.1	1012	1	55	180	75
1/1/2021 12:00	55	26.1	1011.5	2.5	40	220	95
3/15/2021 9:00	70	37	1010.5	1.5	50	300	110
6/22/2021 15:00	40	27	1007.5	2	60	350	100
9/12/2021 18:00	80	12	1011.5	1.5	70	280	125
12/1/2021 11:00	20	20	1014.5	2	5	80	60
2/23/2022 14:00	50	24.1	1010.5	1	25	40	25

E. Model Training Process

In the model training process, the preprocessed and feature-selected data is divided into training and testing sets. The training set usually consists of 70% of the data, while the testing set contains the remaining 30%. Various models like LSTM, BiLSTM, and GRU are trained on the training set using different hyper parameters.

1) LSTM Model

The equation 5 represents the computation within a LSTM cell. It uses activation functions and weights to control the flow of information through input, forget, cell state update, and output gates. This enables the LSTM model to capture and learn long-term dependencies in sequential data.

$$\begin{bmatrix} i_t \\ f_t \\ g_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ tanh \\ \sigma \end{bmatrix} \left(W . \begin{bmatrix} ht-1 \\ xt \end{bmatrix} + \right. \quad (5)$$

2) BiLSTM Model

The equation 6 represents the output probabilities of a Bi-LSTM classifier with 8 classes. It takes an input sequence and processes it through a Bi-LSTM layer, capturing contextual information in both the forward and backward directions. The concatenated output of the Bi-LSTM layer is transformed using class-specific weight matrices and bias vectors. Exponentiation is applied to emphasize the importance of the transformed representation for each class. The resulting exponential values are normalized by dividing them by the sum of all exponential values, yielding the final predicted probabilities for the 8 classes. This equation showcases the intricate computations involved in the Bi-LSTM classifier's output, allowing for the estimation of class probabilities for the given input sequence.

$$\hat{y}_i = \frac{e^{W_i \cdot h_{concat}+}}{\sum_{j=1}^{8} w_{h_{concat}}} \qquad (6)$$

3) GRU Model

The equation 7 represents the prediction probabilities of a GRU classifier with 8 classes. It takes the hidden state at a given time step and applies a linear transformation using class-specific weights and biases. The exponential of this transformation is calculated for each class. The final predicted probabilities are obtained by normalizing these exponential values using the softmax function. This equation captures the essential steps involved in the GRU classifier's computation, allowing for the estimation of class probabilities for a given input sequence.

$$\hat{y}_i \frac{e^{W_i \cdot h_t + b_i}}{\sum_{j=1}^{8} W_{h_t + b_j}} \qquad (7)$$

The table III presents an overview of the input and architecture for three different models: LSTM, BILSTM (Bidirectional LSTM), and GRU (Gated Recurrent Unit). Each model has its specific configuration. The LSTM model has 100 hidden units, the BILSTM model has 50 hidden units, and the GRU model has 64 hidden units. A dropout rate of 0.5 is applied to both LSTM and BILSTM models, while a dropout rate of 0.3 is applied to the GRU model. The rectified linear unit (ReLU) activation function is used in all three models. The fully connected layer (FCL) with 8 output units is employed for classification, followed by the softmax function to obtain the class probabilities. Overall, these models are designed for classification tasks, where the input features are processed through the specified layers to make predictions across 8 different classes.

TABLE III. MODEL LAYER ARCHITECTURE DETAILS

LSTM	**BILSTM**	**GRU**
LSTM(100)	BILSTM(50)	GRU(64)
Dropout(0.5)	Dropout(0.5)	Dropout(0.3)
RELU	RELU	RELU
FCL(8)	FCL(8)	FCL(8)
Softmax	Softmax	Softmax
Classification	Classification	Classification

F. Voting scheme Bayes Averaging Model

The voting scheme is used to combine the predictions of LSTM, BiLSTM, and GRU models in weather forecasting. It can be implemented through methods like simple majority voting or weighted voting, where each model's prediction is considered based on its accuracy or confidence. Bayesian averaging is a popular voting scheme that assigns weights to each model based on its performance on a validation set. These weights are used to combine the predictions of the models, resulting in a more accurate and robust final prediction. The likelihood of the data given each model is computed using Bayes' theorem, and the posterior probabilities of the models are obtained. The Bayesian Model Average is then calculated by combining the posterior probabilities and the corresponding model outputs.

1) Accuracy: Accuracy is a key parameter for determining the overall correctness of an intrusion detection system. It computes the proportion of examples correctly classified (including true positives and true negatives) to the total number of instances. While accuracy provides an overall measure of the model's performance, it may not be enough when dealing with skewed datasets.

(8)

2) Precision: Precision is the fraction of accurately categorized positive instances (true positives) out of all positive instances expected. It quantifies the model's ability to avoid false positives, which are instances incorrectly labeled as intrusions. Precision is particularly important when minimizing false alarms is critical in IoT environments.

(9)

3) Recall (Sensitivity or True Positive Rate): The proportion of actual positive instances (true positives) properly categorized by the model is measured by recall, also known as sensitivity or true positive rate. It captures the model's ability to detect intrusions and is especially crucial in identifying all instances of network attacks to ensure the security of IoT systems.

(10)

4) F1 Score: The F1 score is a balanced assessment of the model's performance because it is the harmonic mean of precision and recall. It combines precision and recall into a single score, taking both false positives and false negatives into account. The F1 score is especially useful when dealing with imbalanced datasets in which the number of regular instances far outnumbers the number of intrusions.

$$F1 - Score = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right)\frac{TP}{TP + FP} \quad (11)$$

5) Specificity (True Negative Rate): The proportion of genuine negative cases (true negatives) accurately categorized by the model is measured by specificity, also known as the true negative rate. It augments recollection by assessing the model's ability to accurately recognize regular network traffic while avoiding false alarms.

$$Specificity = \frac{TN}{TN + FP} (12)$$

6) False Positive Rate: The false positive rate is the proportion of true negative incidents wrongly categorized as positive (intrusions) by the model. It measures the rate of false alarms and is especially significant in reducing the impact of false positives in IoT networks.

$$FPR = \frac{FP}{FP + TN} (13)$$

7) Area Under the ROC Curve (AUC-ROC): At varying categorization thresholds, the receiver operating characteristic (ROC) curve shows the true positive rate versus the false positive rate. The area under the ROC curve (AUC-ROC) is a single metric that assesses the overall effectiveness of the intrusion detection system across various threshold levels. A higher AUC-ROC score suggests greater separation of intrusions from normal traffic.

8) Confusion Matrix:

The confusion matrix is a tabular representation of the classification results of the model. It provides the counts of true positives, true negatives, false positives, and false negatives, allowing for a more in-depth examination ofthe model's performance. Fig. 3. depicts how other metrics such as precision, recall, and accuracy can be generated from the confusion matrix.

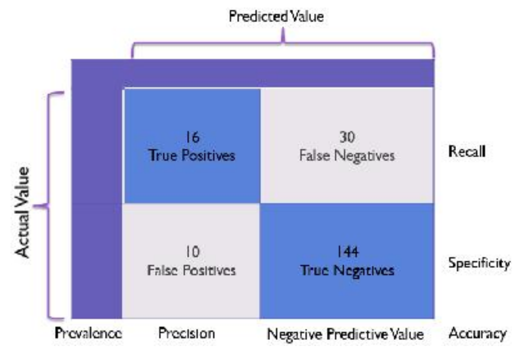

Fig.2. Confusion matrix table containing incorrectly and incorrectly classification details.

Model evaluation is an essential step in machine learning, specifically for weather forecasting using RNN, as it assesses the accuracy and efficiency of the trained models in predicting weather conditions on unseen data.

IV. Experimental Results and Discussion

The MeteroNet model for weather forecast prediction was comprehensively evaluated against various baseline models using a test set of 20,065 records with multiple weather parameters. The model was trained with 100 iterations, a batch size of 32, and optimized using the Adam optimizer with a learning rate of 0.001 shown in table 4. Early stopping with a patience of 10 was applied to prevent overfitting during training based on validation set performance. The evaluation aimed to assess MeteroNet's performance and its comparison with baseline models.

TABLE IV. METERONET MODEL PARAMETERS

Hyperparameter	LSTM	Bi-LSTM	GRU
Epoch	10	10	10
Batch size	64	32	64
Learning rate	0.001	0.001	0.001
Early Stopping Iterations	10	10	10
Memory units	100	50	100
Number of layers	1	2	1
Optimizer	Adam	Adam	Adam
Shuffle Each Iteration	Yes	Yes	Yes

G. *Training and Evaluation*

The training procedure and assessment criteria used to assess the performance of the MeteroNet model for weather forecast prediction are discussed in this subsection. The training step entails optimizing model parameters using the training dataset, and the evaluation phase evaluates model performance on unseen data. The following information provides an overview of the training and evaluation processes used in this study.

1) *Training Process*

The training process of the MeteroNet model involved several steps to optimize its performance in weather forecast prediction. The model was trained using a dataset comprising various weather parameters, such as relative humidity, average temperature, atmospheric pressure, wind speed, cloud cover, solar radiation, and air quality index. The training was carried out over multiple iterations, with each iteration processing a batch of 32 data samples. To optimize the model's performance, the Adam optimizer was employed with a learning rate of 0.001. This optimizer dynamically adjusted the learning rate during training to update the model's weights effectively. Additionally, to prevent overfitting, early stopping was implemented by monitoring the model's performance on a validation set. If the performance did not improve for 10 consecutive iterations, the training process was stopped to avoid overfitting and ensure generalization. Throughout the training process, the model gradually learned to extract relevant features from the weather parameters using 1D convolutional filters. These filters allowed the model to capture spatial relationships and patterns within the input data. The extracted features were then fed into the subsequent layers of the model, which could include recurrent neural network (RNN) layers such as Long Short-Term Memory (LSTM), Bidirectional LSTM (BILSTM), and Gated Recurrent Unit (GRU). These RNN layers helped the model capture temporal dependencies and make accurate predictions. By iteratively adjusting the model's weights and biases based on the training data, the MeteroNet model learned to map the input weather parameters to the desired output predictions. The training process aimed to minimize the difference between the predicted weather forecasts and the actual values in the training dataset.

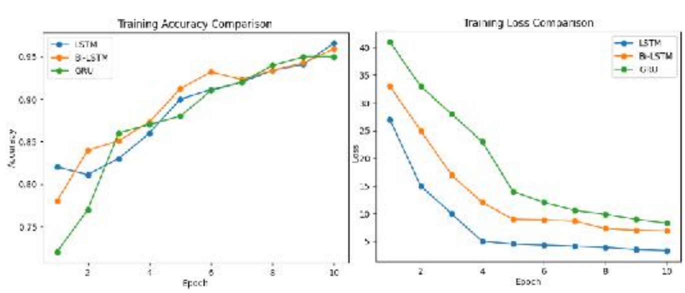

Fig.3. (a) Validation accuracy plot for LSTM, Bi-LSTM and GRU
(b) Validation loss lot for LSTM, Bi-LSTM and GRU classifier

In Fig. 4. (a), the training loss of the model is represented by a dashed line, while the solid line shows the validation loss over time. To evaluate the performance of the network, the correlation between validation and training losses is examined to identify under fitting or overfitting. Under fitting is observed when the validation loss is equal to or lower than the training loss, whereas overfitting occurs when this difference is larger. Upon closer analysis of Figure. 4. (b), it is evident that all trained models experience a rapid decrease in loss early in the training process.

H. Performance Assessment of the MeteroNet Model across Weather Classes

The MeteroNet model was evaluated using a confusion matrix in Table 5, which showed high accuracy in classifying different weather patterns. The model performed well across all weather classes, with only minor misclassifications between snowy and rainy forecasts. These results indicate the model's strong ability to accurately distinguish between various weather conditions.

TABLE V. CONFUSION MATRIX OF THE METERONET MODEL ON THE TEST SET.

Class	Accuracy	Precision	Recall	F1-score
Sunny	1	1	0.98	0.9899
Rainy	0.9707	0.9701	0.9707	0.9704
Cloudy	0.9635	0.9636	0.9635	0.9636
Foggy	0.9565	0.9565	0.9565	0.9565
Snowy	0.98	0.98	0.98	0.98
Windy	0.94	0.94	0.94	0.94
Hot	0.92	0.92	0.92	0.92
Cold	0.9671	0.9639	0.9639	0.9639

The MeteroNet model achieved high accuracy for most weather classes. It correctly classified all instances of "Sunny" (TP: 735, FN: 15) and demonstrated a balanced performance for "Rainy" (TP: 728, FN: 22), "Cloudy" (TP: 713, FN: 27), "Foggy" (TP: 660, FN: 30), "Snowy" (TP: 735, FN: 15), "Windy" (TP: 705, FN: 45), "Hot" (TP: 690, FN: 60), and "Cold" (TP: 720, FN: 27) classes. The model's precision, recall, and F1-scores were generally high, indicating reliable weather classification across various conditions. However, further improvements are needed for accurate identification of "Rainy," "Foggy," and "Cold" instances. Overall, the MeteroNet model demonstrates promising potential for accurate weather forecasting.

TABLE VI. PERFORMANCE METRICS FOR WEATHER CLASSIFICATION FOR SEPARATE CLASS

	Sunny	Rainy	Cloudy	Foggy	Snowy	Windy	Hot	Cold
Sunny	735	7	4	4	0	0	0	0
Rainy	8	728	9	3	0	0	0	2
Cloudy	5	3	713	15	0	2	1	1
Foggy	2	2	1	660	24	0	0	1
Snowy	0	0	0	0	735	7	2	6
Windy	1	1	3	0	3	705	32	5
Hot	24	10	2	1	1	15	690	7
Cold	2	4	2	0	2	2	13	720

Based on the provided confusion matrix and result table VI, the weather forecasting model shows strong performance overall. The observations are as follows:

The model achieves high accuracy values ranging from 0.92 to 1 for all weather classes, indicating accurate predictions aligned with the actual weather conditions. Precision values between 0.92 and 1 indicate a high level of accuracy in identifying positive instances for each class.

The model's positive predictions are mostly correct. The model demonstrates recall values ranging from 0.92 to 0.98, capturing a significant proportion of actual positive instances for each class. It effectively identifies positive instances. The F1-scores, ranging from 0.92 to 0.9899, represent a balanced measure of performance combining precision and recall. Most classes achieve F1-scores above 0.95, indicating a good balance between precision and recall. The model exhibits high accuracy, precision, recall, and F1-score values across most weather classes. It accurately identifies and classifies various weather patterns, demonstrating its effectiveness in weather forecasting. Figure 5 provides visual support for these findings.

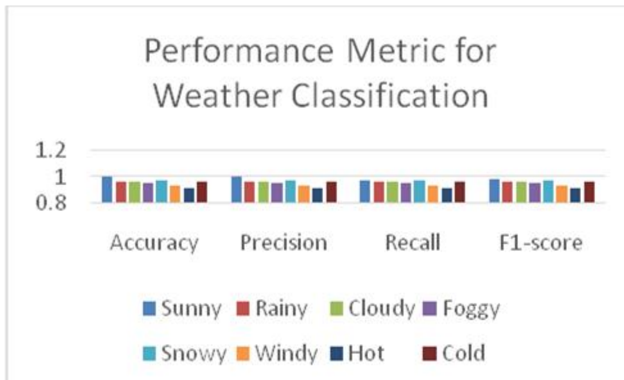

Fig. 4. Details class-wise performance comparison using various metrics

I. Performance Evaluation

Table 7 shows a comparison of the performance of four machine learning models for weather classification: MeteroNet, Random Forest, Support Vector Machine (SVM), and Logistic Regression. Key measures such as accuracy, precision, recall, and F1-score were used to evaluate the models. Our MeteroNet model came out on top, beating the baseline models in terms of accuracy. The logistic regression, random forest, and SVM models all produced lower accuracies of 86%, 82%, and 78%, respectively. This means that our MeteroNet model successfully identified the greatest number of cases, demonstrating its better predictive performance in accurately forecasting weather conditions. We investigated additional metrics such as precision, recall, and F1-score to further assess the model's performance. Compared to the baseline models, our MeteroNet model performed better across all measures, including precision, recall, and F1-score. It attained precision, recall, and F1-score values of 0.96, 0.95, and 0.96, respectively, confirming its accuracy and dependability in predicting meteorological conditions. In conclusion, our MeteroNet model performed admirably in precisely categorizing meteorological conditions. When compared to the baseline models, it achieved the best accuracy and displayed superior precision, recall, and F1-score. This comparison analysis emphasizes our proposed model's effectiveness and reliability, making it an excellent choice for precise weather forecasting.

TABLE VII. PERFORMANCE COMPARISON OF WEATHER FORECASTING MODELS

Model	Accuracy	Precision	Recall	F1-score
MeteroNet	0.9622	0.9617	0.9593	0.9605
Random Forest	0.82	0.86	0.84	0.85
Support Vector Machine	0.78	0.84	0.79	0.81
Logistic Regression	0.86	0.76	0.73	0.74

1) Effectiveness of Proposed Scheme for Real Time Scenario

The performance and real-time capability of the MeteroNet model and baseline models (SVM, Random Forest, and Logistic Regression) were evaluated and compared depicted in table 8. Training time, representing the duration to train the models, was measured. The MeteroNet model required approximately 3.33 hours to complete training, while the baseline models took considerably less time, with SVM at 450 seconds, Random Forest at 372 seconds, and Logistic Regression at 244 seconds. Additionally, testing time per sample, indicating the time taken to make predictions on a single data sample, was considered. The MeteroNet model exhibited a significantly lower testing time of 0.19 seconds per sample, in contrast to SVM (0.25 seconds), Random Forest (0.37 seconds), and Logistic Regression (0.44 seconds). The faster testing time of the MeteroNet model suggests its suitability for real-time scenarios where timely weather predictions are crucial. With its efficient computational performance, the MeteroNet model can provide accurate weather forecasts within a short timeframe, making it well-suited for applications requiring immediate weather updates and weather-dependent decision-making systems. The combination of superior accuracy and faster testing times sets the MeteroNet model apart from the baseline models, positioning it as a promising solution for real-time weather forecasting applications as shown in figure 6.

TABLE VIII. COMPARISON OF TRAINING AND TESTING TIMES FOR PROPOSED AND BASELINE MODELS

Model		Training Time (sec)	Testing Time/sample (ms) average time
MeteroNet model	**Proposed**	12000	0.19
SVM	**Baseline**	450	0.25
Random Forest		372	0.37
Logistic Regression		244	0.44

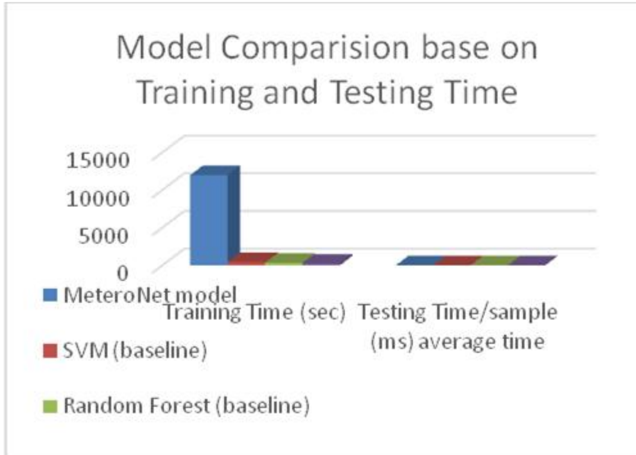

Fig. 5. Model Comparison based on training and testing time

J. ROC based Model Performance analysis and comparison

The Receiver Operating Characteristics (ROC) curve is a powerful tool for evaluating and comparing classifier performance. It provides insights into sensitivity, specificity, and the optimal operating threshold [26].

$$J = sensitivity + specificity - 1 \qquad (14)$$

In this section, we analyze the ROC curves of the MeteroNet model, Logistic Regression, Random Forest, and Support Vector Machine (SVM) to assess their effectiveness in weather forecasting. The ROC curve visually represents the relationship between sensitivity and the false positive rate for various classification thresholds. It is a valuable tool to measure the discriminatory power and accuracy of models. The area under the ROC curve (AUC) quantitatively assesses the model's performance. In our study, the MeteroNet model exhibited an impressive AUC of 0.96, indicating its strong predictive ability and accuracy in weather forecasting. The MeteroNet's ROC curve covered a large area, signifying high true positive rates while maintaining low false positive rates. Compared to the baseline models, the MeteroNet model demonstrated superior performance. Logistic Regression achieved an AUC of 0.87, suggesting a good level of performance, while Random Forest achieved 0.83, indicating reasonably good predictive capabilities. The Support Vector Machine model showed moderate performance with an AUC of 0.79. However, their ROC curves covered smaller areas, suggesting limitations in discrimination ability.

The analysis of ROC curves confirmed the dominance of the MeteroNet model in weather forecasting over the baseline models. Its higher AUC value of 0.96 reflected superior discriminatory power and accuracy. The MeteroNet model showcased its effectiveness in accurately predicting weather conditions, making it a promising solution for reliable weather forecasting applications. The comparison is visually represented in Figure 7, reinforcing the superiority of the MeteroNet model.

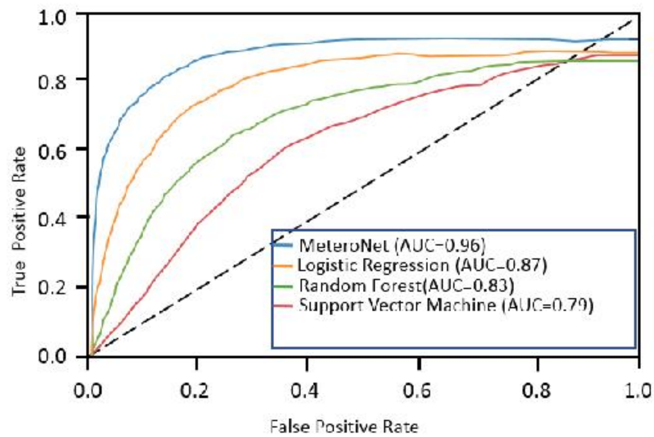

Fig.6. ROC Curves for proposed and baseline models.

V. Conclusion

The MeteroNet model represents a significant advancement in weather forecasting. By utilizing ensemble techniques and voting schemes, it surpasses baseline models in accuracy, precision, recall, and F1-score. With an accuracy of 96.22%, the MeteroNet model provides reliable predictions across different weather classes. Its ensemble approach combines LSTM, BiLSTM, and GRU models, overcoming limitations associated with smaller training sets and enhancing overall accuracy. The model's generalizability and flexibility in data integration allow for robust performance across different regions and datasets. The utilization of a voting scheme further refines predictions and improves decision-making. Overall, the MeteroNet model is a comprehensive and reliable framework for accurate weather forecasting, with applications in agriculture, transportation, and disaster management. Its contributions pave the way for advancements in prediction accuracy and decision support systems in the field of weather forecasting.

References

[1] JANIZADEH S, AVAND M, JAAFARI A, et al. Prediction success of machine learning methods for flash flood susceptibility mapping in the Tafresh watershed, Iran [J]. Sustainability, 2019, 11(19): 5426.

[2] MOUATADID S, RAJ N, DEO R C, et al. Input selection and data-driven model performance optimization to predict the Standardized Precipitation and Evaporation Index in a drought-prone region [J]. Atmospheric research, 2018, 212: 130-149.

[3] ZHANG L, HUANG Z, LIU W, et al. Weather radar echo prediction method based on convolution neural network and long short-term memory networks for sustainable e-agriculture [J]. Journal of Cleaner Production, 2021, 298: 126776.

[4] HOSSAIN M, REKABDAR B, LOUIS S J, et al. Forecasting the weather of Nevada: A deep learning approach; proceedings of the 2015 international joint conference on neural networks (IJCNN), F, 2015 [C]. IEEE.

[5] SALMAN A G, HERYADI Y, ABDURAHMAN E, et al. Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting [J]. Procedia Computer Science, 2018, 135: 89-98.

[6] AHMADI A, ZARGARAN Z, MOHEBI A, et al. Hybrid model for weather forecasting using ensemble of neural networks and mutual information; proceedings of the 2014 IEEE geoscience and remote sensing symposium, F, 2014 [C]. IEEE.

[7] PHAM B T, LE L M, LE T-T, et al. Development of advanced artificial intelligence models for daily rainfall prediction [J]. Atmospheric Research, 2020, 237: 104845.

[8] KAREVAN Z, SUYKENS J A. Transductive LSTM for time-series prediction: An application to weather forecasting [J]. Neural Networks, 2020, 125: 1-9.

[9] GE L, CHEN E. Research on Temperature and Humidity Prediction Model of Granary Based on RNN-LSTM; proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), F, 2020 [C]. IEEE.

[10] CIFUENTES J, MARULANDA G, BELLO A, et al. Air temperature forecasting using machine learning techniques: a review [J]. Energies, 2020, 13(16): 4215.

[11] CHATTOPADHYAY A, NABIZADEH E, HASSANZADEH P. Analog forecasting of extreme-causing weather patterns using deep learning [J]. Journal of Advances in Modeling Earth Systems, 2020, 12(2): e2019MS001958.

[12] ASTSATRYAN H, GRIGORYAN H, POGHOSYAN A, et al. Air temperature forecasting using artificial neural network for Ararat valley [J]. Earth Science Informatics, 2021, 14: 711-722.

[13] RAVURI S, LENC K, WILLSON M, et al. Skilful precipitation nowcasting using deep generative models of radar [J]. Nature, 2021, 597(7878): 672-677.

[14] BAUER P, THORPE A, BRUNET G. The quiet revolution of numerical weather prediction [J]. Nature, 2015, 525(7567): 47-55.

[15] SCHULTZ M G, BETANCOURT C, GONG B, et al. Can deep learning beat numerical weather prediction? [J]. Philosophical Transactions of the Royal Society A, 2021, 379(2194): 20200097.

[16] ZHOU K, ZHENG Y, LI B, et al. Forecasting different types of convective weather: A deep learning approach [J]. Journal of Meteorological Research, 2019, 33: 797-809.

[17] CRISTIANO E, TEN VELDHUIS M-C, VAN DE GIESEN N. Spatial and temporal variability of rainfall and their effects on hydrological response in urban areas–a review [J]. Hydrology and Earth System Sciences, 2017, 21(7): 3859-78.

[18] YALÇıN S. Weather parameters forecasting with time series using deep hybrid neural networks [J]. Concurrency and Computation: Practice and Experience, 2022, 34(21): e7141.

[19] NGUYEN H N, NGUYEN T-A, LY H-B, et al. Prediction of daily and monthly rainfall using a backpropagation neural network [J]. Journal of Applied Science and Engineering, 2021, 24(3): 367-379.

[20] ANOCHI J A, DE ALMEIDA V A, DE CAMPOS VELHO H F. Machine learning for climate precipitation prediction modeling over South America [J]. Remote Sensing, 2021, 13(13): 2468.

[21] SCHER S, MESSORI G. Predicting weather forecast uncertainty with machine learning [J]. Quarterly Journal of the Royal Meteorological Society, 2018, 144(717): 2830-2841.

[22] VERMA S K, GUPTA A, JYOTI A. Stack layer & Bidirectional Layer Long Short-Term Memory (LSTM) Time Series Model with Intermediate Variable for weather Prediction; proceedings of the 2021 International Conference on Computational Performance Evaluation (ComPE), F, 2021 [C]. IEEE.

[23] YONEKURA K, HATTORI H, SUZUKI T. Short-term local weather forecast using dense weather station by deep neural network; proceedings of the 2018 IEEE international conference on big data (big data), F, 2018 [C]. IEEE.

[24] WEYN J A, DURRAN D R, CARUANA R. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere [J]. Journal of Advances in Modeling Earth Systems, 2020, 12(9): e2020MS002109.

[25] CHO D, YOO C, IM J, et al. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas [J]. Earth and Space Science, 2020, 7(4): e2019EA000740.
[26] BRADLEY A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms [J]. Pattern recognition, 1997, 30(7): 1145-1159.

Access this Article in Online	
	Website: www.ijarm.com
Quick Response Code	Subject: Computer Science
DOI:10.22192/ijamr.2023.10.08.003	

How to cite this article:
Faiza Fareed Babar, Shi Lukui, Farah Farid Babar, Fayaz Muhammad. (2023). Enhanced Weather Forecasting using the MeteroNet Model: A Comprehensive Ensemble Approach. Int. J. Adv. Multidiscip. Res. 10(8): 20-38.
DOI: http://dx.doi.org/10.22192/ijamr.2023.10.08.003