Journal of Information Systems Engineering and Management
2025, 10(15s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Improved Hybrid_Stacked Deep Neural Network (HDNN) Model for Enhanced Weather Forecasting

Umesh Kumar[1*], Dr. Neetu Sharma[2]
[1]*Government Mahila Engineering college Ajmer, Rajasthan, India, E-mail:umesh@gweca.ac.in*
[2]*Government Engineering college Ajmer, Rajasthan, India, E-mail:drneetu@ecajmer.ac.in*
* Corresponding Author: umesh@gweca.ac.in

ARTICLE INFO

Received: 28 Nov 2024

Revised: 22 Jan 2025

Accepted: 01 Feb 2025

ABSTRACT

Change in climatic conditions is considerably one of the most serious topic confronting the globe today. Weather predictions are based on various temporal and spatial scales along with chaotic dynamics with very high dimensionality domination, which becomes a cause for multiple complex problems in the field. The cutting-edge numerical models with high computational cost are not sufficient for several applications and hence it calls for expansion of work by using Artificial Intelligence to deal with such problems. The current work will look into the possibility of forecasting weather characteristics utilizing the Deep Neural Network (DNN) models. The aim is to find out the capability of DNN in predicting weather conditions. The proposed multiple input single output (MISO) regression model is explored by using well established DNN approaches like Long Short-Term Memory (LSTM) along with the well established Bi-directional LSTM(BiLSTM). Historical weather station data of ten years is being used for this research and also for the purpose of model training. It has been pre-processed to obtain accurate data in desired format. For accurate weather forecasting, the proposed model has also tested utilizing various DL settings and controls and then after performance evaluation is done using different regression metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Square Error (MSE) and compared with the well-established statistical models like ARIMA, CRNN, LSTM and Bi-LSTM, which were altered as per suitability. To determine the accurate weather forecasting model, comparison research of existing models and proposed weather forecasting model is conducted.

Keywords: Weather Forecasting, ARIMA, CRNN DNN, MISO, LSTM, BiLSTM

I. INTRODUCTION

In earlier days, observing the sky was the first step in forecasting, which was supplemented by the invention of various instruments for weather prediction like Anemometer, Hygrometer and Thermometer etc. [1]. In recent years, forecasting tools and observatory methods have advanced to the highest level, and the introduction of specialized meteorological satellites [2] and radars has made it feasible to keep a close and precise eye on the weather. In today's fast telecommunication network, countries exchange weather observations and updates rapidly through the help of Meteorological satellites to produce near accurate predictions[3]. Aside from different public sector agencies and weather observatory stations, a number of private organizations have developed the ability to forecast the weatheristic parameters. The dissemination of this information via the newest smart devices is a good indicator of the expansion and advancement of weather forecasting and its technologies. The prime goal of weather forecasting is to provide information about the expected weather conditions, such as temperature, precipitation, wind, humidity, and cloud cover, to help people make informed decisions.

Since the early days of meteorology, when it was primarily based on observations of the current weather conditions and straightforward rules of thumb, weather forecasting has advanced significantly. Today, weather forecasting is a sophisticated and complex science that uses a vast network of weather monitoring stations, cutting-edge computer models, and remote sensing technologies. As a result, depending upon the number of factors, the procedure of weather related predictions becomes complex and challenging as well [4]. Fluctuations in weather conditions are noticed every few hours and extreme changes occur from time to time [5]. Being aware about the weather conditions earlier itself leads to reduction in the losses and helps us in numerous ways. Weather forecasting has wider applications varying from being useful for a student to keep an umbrella when being aware that it would rain in coming time to being useful for governmental establishments in emptying a locality when being aware about the possibility of heavy rain in that area. Forecasting is the undertaking of expectation of the environment at a future time and a given zone[6]. In the early days, this has been completed through physical conditions in which the air is considered as runny. While the present condition of the earth is analyzed and future projections are made by mathematically addressing those circumstances we cannot accurately forecast the weather beyond a few days, though advancements in science and technology may enhance this capability. Although the current state of the earth is examined and the future is predicted by mathematically resolving those conditions, we are unable to determine the precise climate for more than a few days, and this can be improved with the help of science and innovation [7]. The use of weather forecasts is widespread, from public safety and disaster management to agriculture and transportation. People can make decisions that prevent damage to property, save lives, and enhance quality of life with the use of accurate and timely weather information. The traditionally used weather forecast procedures that used satellite images and weather stations are costly due to the inclusion of processes being high in cost and complexity both [8]. Weather forecast by the use of Machine Learning (ML) is low in cost, takes lesser time, higher in convenience, real in time and precise in nature [9]. A few of the present researches related to weather forecasting including ML technique involved the usage of much of the former weather data [10]. The accuracy of the forecasts depends upon the models being trained with. Thus, it becomes much essential for any ML model to be trained with a highly precise data. The data attained from a number of sources is not trustworthy all the time. Thus, it becomes necessary to preprocess the data. Preprocessing the data include removing unnecessary columns that are irrelevant to the model's prediction, eliminating zero values, combining the identical columns, and a number of other pre-processing steps [11].

II. LITERATURE SURVEY

This review of literature is an evaluative report of facts and figures drawn from the literature in relation with the Weather Forecasting and DL models. The literature is described, summarized, evaluated, substantiated, and clarified with the aid of these reviews, which also give the study a theoretical base and aid in defining the scope. It is a collection of scholarly works that provides a summary of current knowledge along with relevant research findings and theoretical as well as methodological contributions. Numerous designs, models, simulated systems, and prototypes have been developed by meteorologists, scientists, and researchers to anticipate weather factors with a high degree of accuracy.

A hybrid model was employed by Liu [12] to estimate wind speed. The original datasets were transformed into a variety of different sub-series once the EMD approach was used. Once a predictive model is created, each sub-series' multiple-step prediction was made using ANN. To arrive at the final prediction of wind speed, all the predicted results were pooled in the sub-series. ANN and ARIMA model, both had a bearing on the performance of the hybrid model. The results for wind speed suggested that the hybrid model's accuracy was satisfactory and practical for handling non-stationary time series data. Same year a hybrid model developed from the ARIMA-ANN and ARIMA-Kalman processes was put forth by Authors[13] in 2012 to predict wind speed. For a portion of the wind speed sample, an ANN model was created. To determine the appropriate structure for this modelling method, a time series ARIMA was used. For the identical section of the wind speed data, a Kalman model was created. To find the ideal Kalman model parameters, a time series ARIMA was used.

Author [14] employed a different strategy to forecast the value of a single variable in the future, using many features like pressure, temperature, humidity, precipitation, wind and moisture. Many ML and DL algorithms, including TCN [15], LSTM with multiple-input multiple-output (MIMO), and multiple-input single-output (MISO) approaches, were used to carry out this. Implementing the Jordan Pi-Sigma Neural Network (JPSN) for time series data, developed by Authors [16], was an innovative strategy. In order to estimate the temperature, they merged the Jordan Neural Network and Pi-Sigma Neural Network techniques in this paper. The model's MSE is impressively low, but it fails to meet the criterion outlined by authors [17], and its performance is only acceptable if the NMSE is less than 0.5.

In contrast to additive hybrid models, author's [18] assumption was that the time series data was the multiplication of a nonlinear and a linear series. Their hybrid model calculates an ARIMA forecast by fitting the ARIMA model to the provided time series data. The residual error series is then obtained by eliminating ARIMA projections from the original series. The generated residual error series is considered nonlinear and modeled using ANN to obtain forecasts. To obtain the final estimates, ANN forecasts are multiplied by ARIMA forecasts. This model outperformed the other models in three-time series in terms of forecast accuracy. The method, however, is inapplicable to a series with zero ARIMA predictions.

Zaytar presented a DNN architecture for time series weather prediction [19]. Sequences of weather values for the short-term forecast of temperature, humidity, and wind speed data were mapped using several stacked LSTMs. The results reveal that the suggested model was competitive and judged to be a better alternative when compared to standard methods for the forecasting of general weather conditions. Hourly meteorological data spanning 15 years, from 2000 to 2015, were utilized to train the model. DL and RNN were combined by Authors [20] to anticipate wind speed with improved short-term outcomes in comparison to other models. The investigation used time series data from northeastern US windmills, which showed to be excellent in terms of forecasting. The weather forecasting algorithms like Linear Regression, MLR, SVR, and ARIMA were developed by Authors [21]. The predictive ability was calculated using the RMSE parameter, which identifies ARIMA as the top prediction model.

Authors [22] employed an LSTM model variation to predict ground visibility and found that the model's predictive abilities may be improved with the introduction of moderating variables. Based on the results of experiments, the proposed merged-LSTM model had a 4.8% higher improvement in accuracy. For the purpose of forecasting the weather, Authors worked on RNN using LSTM [23]. Numerous meteorological parameters were gathered from NCDC, and using the LSTM technique, NNs were trained for various combinations of weather parameters "temperature, precipitation, wind speed, pressure, and humidity", in order to forecast the future weather condition using LSTM. It was discovered that this method's prediction accuracy outperformed other approaches.

Authors employing a NN [24] was able to anticipate the weather by using data from the past. In particular, a model called CRNN was created based on CNN and RNN, and it makes use of NNs to learn the time and space correlation of temperature changes from historical data. The model was trained using daily temperature data from the Chinese mainland, and the forecasted temperature has an inaccuracy of about 0.907°C. By stacking LSTM layers with different numbers of units in each layer, Author [25] developed a DL approach for the prediction of weather data in 2020. A multiple-input multiple-output (MIMO) structure was used to forecast similar weather parameters for a specific time sequence using various weather variables as input characteristics. Testing was conducted on the resulting models for the prediction of Temperature, humidity and windspeed. Experiments were conducted with a variety of hyper-parameters, including the number of LSTM layers and units and the learning rate. The results revealed that the cascaded models outperform the state-of-the-art LSTM or 1D CNN for shorter period prediction. Author's review [26] found that DL models outperform ANNs in terms of mean square error while predicting air temperature. Authors [27] worked on forecasting the highest temperature using DL. Utilising meta-learning methods for hyper parameter optimization, the DL network structure was improved. Three different models—ANN, RNN, and LSTM—were tested and trained using the same dataset in order to select the best network architecture. The results of the work demonstrate that in case of long-term forecasting, the hybrid model of an LSTM network along with GA outperforms rest of the models. Additionally, Authors [28] claims that DNNs can provide a superior feature space for weather data sets to anticipate weather changes over the next 24 hours.

Ensemble weather forecasting approach is useful for capturing the uncertainty inherent in weather forecasting, and for providing predictions with more accuracy and reliablity. Authors [29] works on "The ECMWF Ensemble Prediction System: Methodology and Validation", which is one of the most commonly used ensemble forecasting systems in the world. which comprises of validation data as well as a description of the approach used to construct the ensemble forecasts. Authors [30] published "Ensemble Forecasting at NCEP and the Breeding Method". The concept of ensemble forecasting was introduced in this study, which involves creating numerous forecasts with modest differences in initial conditions and model parameters. Breeding approach was discussed in the work, which generates these variants through an iterative procedure. Authors [31] presents "Ensemble-based Probabilistic Forecasting at Horns Rev" an in-depth examination of ensemble forecasting in meteorology. In which the history of ensemble forecasting, numerous methods for generating ensembles, and applications of ensemble forecasting in many domains were discussed. Authors [32] worked on "A Hybrid DL Framework for Short-Term Wind Speed Forecasting" to forecast wind speed and this paper presents a hybrid DL architecture that integrates LSTM networks with other ML approaches. The authors train and test their model on historical wind speed data and discover that the LSTM network can capture both short-term and long-term dependencies in the data, resulting in more accurate wind speed forecasts.

Authors [33] combined variety of DL approaches for the forecasting of weather and proposed the Hybrid_ Stacked Bi-LSTM model, which combines both LSTM and Bi-LSTM and aids in the quick prediction of future weather conditions. In order to replicate the entire dynamics of the revised general circulation model, DL models were used, which increased the accuracy and stability of long-term climate time series as well as the outcomes for weather prediction. Authors[34] presents a concise overview of weather and forecasting techniques, including categorisation and general weather forecasting methods, as well as their benefits, drawbacks, and limits, which may be useful to the community of researchers and students.

While ensemble forecasting is computationally intensive, it is worth the effort in order to produce better weather forecasts. To alleviate this, the new FuXi-ENS model [35] provides up to 15-days high-resolution (0.25°) global ensemble weather data. FuXi-ENS is a step in the evolution from past methods of ensemble weather forecasting that just used flow-dependent perturbations, and beats traditional forecasting methods (e.g. ECMWF) in terms of accuracy by using a Variational AutoEncoder (VAE) with more advanced loss functions. In the same year the authors [36] improved flood forecasting by utilizing the "meteo-hydro-AI" technique, which integrates hydrological modeling, weather forecasts, and AI-based bias correction. The methodology, which included a high-resolution land surface model and ECMWF meteorological data, was tested in the Luo River basin (2010–2017) with lead durations of up to seven days. In comparison to the conventional ensemble streamflow prediction (ESP) method, it was able to predict flood hydrographs with success. With an improvement in Nash-Sutcliffe values of 0.27 to 0.82, a decrease in RMSE of 22 to 49%, and increased dependability and discrimination, the results demonstrated enhanced forecast accuracy. Same year the authors [37] present a NN-based post-processing technique for ensemble forecasts that uses a single model for all stations and lead times, overcoming the drawbacks of conventional methods. To model intricate forecast distributions, it integrates normalizing spline flows. The approach demonstrates state-of-the-art performance and beats other models when tested on the EUPPBench standard for 2-meter temperature forecasts in Europe. Probabilistic machine learning-based weather model GenCast can produce high-resolution 15-day global forecasts in just 8 minutes [38]. Trained on four decades' worth of reanalysis data, it beats the ENS system on 97.2% of targets and excels at forecasting wind power, tracking tropical cyclones, and predicting freak weather. This innovation establishes a new benchmark for operational weather forecasting that is both precise and effective

III. METHODOLOGY

The research approach collects pertinent data in a systematic manner to support or refute the work and to control the dissemination of logical conclusions. The overall methodology employed for the suggested task is shown in Figure 1, which illustrates a methodical approach to problem solving. The first stage is to gather weather data, which is initially unclean and can be used only after preprocessing. As a result, preprocessing is needed following data collection for data cleaning, feature selection, and data standardization. A drop in performance level may occur with an increase in the number of variables, despite the fact that many models are capable of training the data connected to the raw input. Therefore, a feature selection is required in order to reduce the number of input variables. A method for choosing the optimal subset of variables that also results in a reduction in the input dimension is feature selection. Finding the highly representative input data is the goal of feature extraction. Following preprocessing, clean data in the desired format is acquired and can then be divided into a set of training and another set of testing data. All the models used in this work are trained using the similar training dataset.

Performance analysis is then carried out by using evaluation metrics like "Mean Absolute Error", "Mean Square Error" and "Root Mean Square Error". The best model for weather forecasting is finally determined by performing a comparison analysis based on the performances of different models.

A. Dataset and Pre-processing

For this work, weather information from Szeged, Hungary, is used. The data collection has 12 columns and 96453 rows, which contains hourly weather data. The attributes in dataset are "Time", "Summary", "Precipitation", "Temperature", "Apparent Temperature", "Humidity", "Wind Speed", "Wind Bearing", "Visibility", "Loud Cover", "Pressure", and "Daily Summary". The data set spans the years 2006 through 2016.

Source: https://www.kaggle.com/budincsevity/szeged-weather

Data Preparation: The obtained meteorological data is primarily prepared to be used in NN models, thus first we must clean and interpolate the data to remove NULL values. Data was often inconsistent, erroneous, and missing values. "Clean" refers to the absence of any unnecessary information in the data, as well as any text, labels, symbols or characters. The data must be cleaned before using data with ML models, also all the NULL values should be eliminated.

After the dataset being linearly interpolated for the inclusion of missing values, the unnecessary columns which has only one or less unique values should be removed, because these parameters will not impact the training process. The next step is to find the correlations between parameters and highly correlated parameters should be removed. After that, all the non-numerical values present in our dataset should be converted to numerical values and for this 'LabelEncoder' is used that can be attained by the use of 'Sklearn' Library. 'Sklearn' comes up with a much efficient tool for the purpose of encoding the levels of categorical features into numerical values. All weather parameters are standardized using Standard Scaler for the testing dataset after completing these steps in order to preserve the values in a scale.

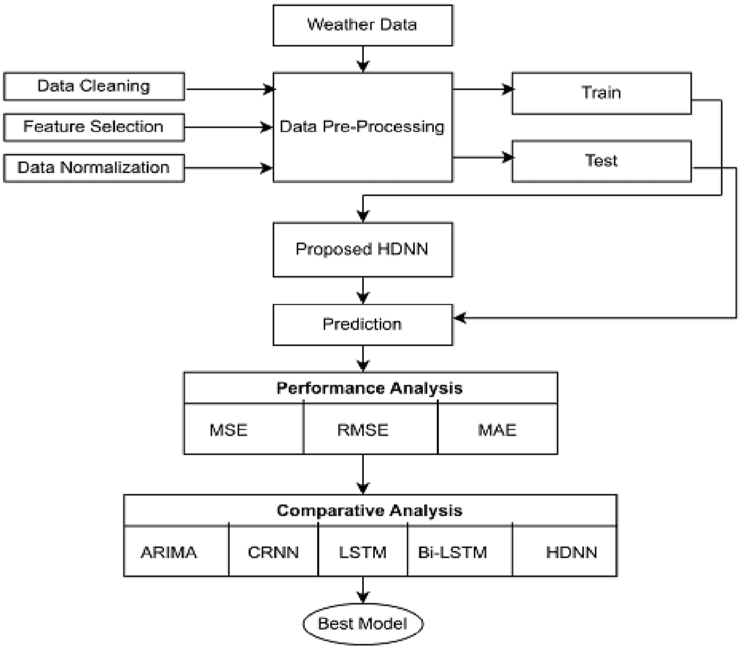

Figure. 1: Methodology

Pertinent studies demonstrate that training neural networks with standardized data is frequently more effective and produces superior predictions.

The standard scaler, a scaling technique, is used to scale the results, which are centred around the mean. As a result, the produced distribution is having a unit standard deviation and the attribute's mean is zero. The standardization formula is as follows.

$$X' = \frac{X-\mu}{\sigma} \tag{3.1}$$

i.e. σ = Standard deviation and μ = Mean of feature values.

This is a essential preprocessing procedure that is carried out generally before training different ML models in order to standardize the functionality range of the input dataset. It is utilized in the Data Preprocessing stage known as Feature Scaling. In essence, it is used to scale the feature's magnitude within a specific range. The data gathered from the real world typically differ greatly from one another and directly affect the performance of the model. Scaling the data before processing it is therefore always recommended. Given N features, Standard Scaler for each value in a particular feature can be calculated by:

$$\frac{\text{Value} - \text{Mean of Feature}}{\text{Standard deviation of Feature}} \tag{3.2}$$

StandardScaler is a Python package that is available to prevent calculations. StandardScaler is often generated using functions like fit_transform(dataset) for any dataset. The fit and transform routines can be used independently. While the convert function evaluates and replaces the values, the fit function computes the mean and standard deviation. Finally, the dataset is transformed into daily data so that the models can be trained.

The NNs provide the predicted values in a normalized manner. De-normalized data are used to compare the projected outcomes with the actual results after these data have been de-normalized and transformed into a human-understandable format.

The daily dataset is used to extract a total of eight important weather factors. The Dataset is used in the ratio of 70 and 30, as 70% of the data is used for model training and the 30% is used for model testing to see which one is the most effective for weather forecasting.

After performing all the necessary preprocessing steps, the details of our final dataset are depicted in figure 2.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4019 entries, 2005-12-31 00:00:00+00:00 to 2016-12-31 00:00:00+00:00
Freq: D
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Summary                 4019 non-null   float64
 1   Precip Type             4019 non-null   float64
 2   Temperature (C)         4019 non-null   float64
 3   Humidity                4019 non-null   float64
 4   Wind Speed (km/h)       4019 non-null   float64
 5   Wind Bearing (degrees)  4019 non-null   float64
 6   Visibility (km)         4019 non-null   float64
 7   Pressure (millibars)    4019 non-null   float64
dtypes: float64(8)
memory usage: 282.6 KB
```

Figure 2: preprocessed dataset

Time-series sequential data include historical weather data. Thus, a fine-grained model for weather forecasting may be developed and evaluated for these data using sequential data modelling techniques.

B. Implementation Setup

To enable DL models and NNs in a computer system certain model libraries like, TensorFlow, BVLC, Theano, and Pfnet and are used. Popular NN models employ these model libraries as their backend. The TensorFlow backend is the most often used. Keras is a more streamlined, high-level Application Programme Interface (API). With a TensorFlow backend, it enables quick prototyping. With its rich APIs and modular design philosophy, Keras enables the construction and testing of NNs while using lesser code. A sample, batches and epochs are set, allowing for the independent processing of data and the identification of recognizable phrases. The Keras open-source neural-network programme is utilized to create the model, and the Keras API is used for this work.

Hardware Used: A general purpose computer is used for each experiment in this work, with an Intel Core i7 CPU which comprises of 4cores, 8 logical processors, and 32 GB of RAM. This CPU operates at 3.4 GHz. These workstations provide 8 GB of (GPU) Graphical Processing Unit memory in addition to other fundamental settings.

C. Model Training

In this process, patterns are identified in the dataset to be used for training, that link the input data features to the target (i.e., the forecast or labeling response) [39]. Model training in supervised learning uses labelled input data to assist establish the right values for all "weights" and "biases" [40]. A number of parameters like learning rate(LR), cost function, epochs, batch size and optimizer are been configured throughout this procedure, along with other deep network-related elements.

Learning Rate: In terms of significance, after LR, network configuration is to be considered [41]. On the basis of the estimated error, model weights are adjusted each time and LR keeps a check on the level of alteration to be made in model. If LR is too little, a long training process is required; nevertheless, if it is too large, the training process appears unstable [42]. As a result, it is extremely difficult to set up the learning rate in an ML. LR-scheduler was developed to resolve this issue, which allows the LR to be adjusted during training after having decreased according to a pre-determined time-table [43]. Scheduler is available in three alternative configurations: exponential decay, which drops LR exponentially for each epoch, step-decay, which drops LR by a factor every few epochs, and time-based decay, which drops LR by a factor every epoch [44].

Optimizer: ML networks frequently employ optimizers to lower a certain cost function after modifying the model's weights and bias settings. Simply by "futzing" with the weights and optimizers, the model is shaped in such a manner that it gets into the most accurate form. The cost function in deep models can be solved using stochastic optimizers like Adam and SGD. Adam optimizer is used in this work.

Epoch: A thorough dataset presentation that needs to be learned throughout the model training process is called an epoch. Learning machines employ a variety of epochs and iterative algorithms, including feed-forward neural networks, throughout the learning phase. As a result, the epoch is a parameter that establishes how frequently the training dataset is utilized to update the weights.

Batch Size and Samples: The sample is one data row from a dataset. Accordingly, a dataset's total sample count (or total number of rows) can be thought of as many samples. The training dataset, in general, contained a large number of samples that must be propagated through before changing the internal model parameters can be specified as the batch-size. One or more batches can be created from the training dataset. These are the most widely used batch sizes: 32, 64, 128, and 256. The training procedure is effective and requires less memory when the sample size is reduced.

Models: Model configuration specifies the number of layers and various node densities in the deep network. There are many models that can be trained using the preprocessed weather data. Five distinct models, namely ARIMA, CRNN, LSTM, Bi-LSTM and the proposed HDNN , have been considered for the current work. The ARIMA model forecasting process differs from other approaches in that it repeatedly selects a suitable model from a set of models rather than taking into account any specific pattern from historical data. The remaining are deep learning models. The data based on time series is used to extract lateral features using one-dimensional convolutional layer. Time series is utilized to extract temporal characteristics using LSTM layers. Combining LSTM with Bi-LSTM results in the proposed HDNN. Three stacked LSTM layers and three bidirectional LSTM layers make up the suggested model. In this work all the models are implemented using similar dataset.

D. Performance Analysis

An important part of the operation of the forecasting system is performance assessment of a number of weather forecasting models. Before selecting the best model for the forecast, each model should be evaluated. Testing dataset is used to evaluate the accuracy and the method for assessing a problem solution is the model performance. In other words, the model is trained and then employed to forecast weather parameters and to determine the model's accuracy by comparing the outcomes with the labels. In regression models, the model accuracy could be expressed numerically, and MSE, MAE, and RMSE are frequent evaluation measures.

IV. PROPOSED HYBRID_STACKED DNN (HDNN)

In the present work HDNN using LSTM and Bi-LSTM neural network structure (Hybrid_Stacked DNN) is proposed as shown in figure 3. LSTM can be effectively applied for the forecasting of weather parameters because while using sequential data it has ability to learn long-term dependencies in a effective manner, which makes them compatible for time series forecasting tasks such as weather predictions. Bi-LSTM is a kind of NN that can be used for the forecasting purpose, similar to LSTM. Bi-LSTM models have the added benefit of being able to capture dependencies in both forward and backward directions of sequential data.

The proposed MISO based DL model with stacked LSTM and Bi-LSTM layers is shown in figure 3, Three stacked LSTM layers and three stacked Bi-LSTM layers make up the overall composition, or nodes, of the proposed HDNN structure. These nodes using a set of coefficients and use data and input in a effective combination to do the computations. The experimental results are used to determine the ideal quantity of layers and required memory cells in each layer. These models might discover long-term dependencies by merging memory units. The network updates previously hidden states, forget previously hidden states, and learn new information with the help of these memory units.

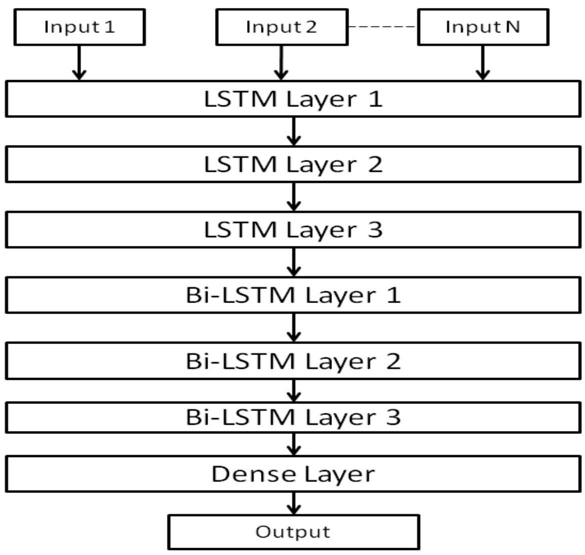

Figure 3: Proposed Hybrid_Stacked DNN

The overall structure consists of an Input Layer, three LSTM layers of 256, 128 and 64 units respectively, three stacked Bi-LSTM layers of 32, 16, 8 units respectively, one dense layer of unit 8 and one output layer. An issue of overfiting of training data may occur with LSTMs and Bi-LSTMs, reducing their predictive skill so in order to address this issue a dropout layer is also incorporated for every LSTM and Bi-LSTM layer with value of 0.2 for reducing over fitting and improving model performance. To optimize our model, 'Adam' is used as optimizer. LSTM models are required to train with larger datasets, and even with the use of Graphical Processing Unit, this procedure frequently takes several days.

The suggested MISO model is a regression model that receives numerous input parameters and produces a single parameter. In section 3.3, the rationale for using multiple input, single output regression models is discussed. Seven surface weather parameters are sent into the network as inputs for this model variance, as was mentioned in Section 3.1, and it is anticipated that it will forecast one suitable parameter as the output. In this method, four distinct models are needed to forecast the weather because each one is trained to predict a specific meteorological parameter.

HDNN is proposed to choose the best DL model with the lowest MSE, several configurations and controls are investigated. Every network configuration consistof unique number of layers, each one of them is made up of unique number of nodes. These setups have tested a variety of parameters, primarily learning rate and optimizer. With Adam optimizer, batch size 32 and learning rate 0.01, the model yields the best value. Due to the fact that the present work is based on regression modelling, which is derived using Equation 3.3, MSE cost function is chosen to determine the loss for experiments. The 177 epochs for temperature, 345 epochs for wind speed, 179 epochs for humidity, and 243 epochs for rainfall parameter had the lowest MSE. The training dataset's shape is (2792, 30, 7), wherein 2792 is the number of samples, followed by time steps and data_dim. This hyperparameter indicates the number of parameters in each timeslot.

LSTM and Bi-LSTM network with similar configuration and controls, individually generated prediction values but with lower accuracy for all the four parameters. It drove the way forward for the designing of hybrid model assuming that HDNN model improves the prediction results.

V. RESULTS

The HDNN model is trained with the training dataset. The expected outcomes and the actual data are compared for each of the four parameters. In order to find the model with the lowest mean square error (MSE), a range of controls and configurations are employed during model testing. The "save the best model" technique is applied during model training. In this instance, the system uses the stored model to validate the loss function value for each epoch. The system only deemed the new model to be the best model if its loss was less than that of the model that had been saved earlier.

Table 1: Evaluation results for Hybrid_Stacked DNN model

Parameters	Evaluation Metrics		
	MSE	RMSE	MAE
Temperature	0.486	0.697	0.517
Wind Speed	0.856	0.925	0.677
Humidity	0.0002	0.0155	0.0119
Rainfall	0.763	0.873	0.290

The testing dataset's prediction is obtained using the saved best model, and the accuracy of the findings is assessed in relation to the source data. The MSE measure is mostly used to evaluate the model. MAE and RMSE are also used while analyzing the data. The calculation of the error is the same for each of these evaluation metrics. For the model, these error levels should be as low as possible to operate more effectively. The obtained MSE, RMSE, and MAE values of HDNN are shown in Table 1 for each of the four parameters.

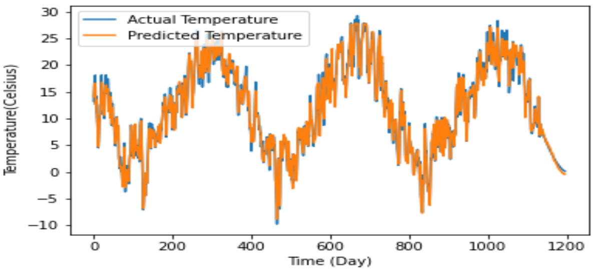

Figure 4: Actual v/s predicted Temperature for HDNN model

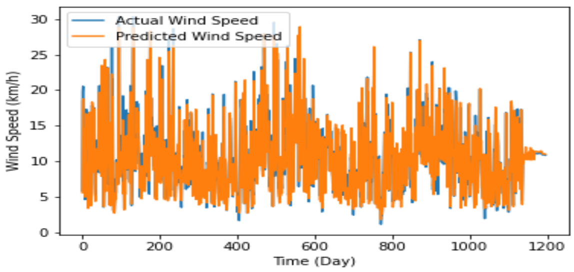

Figure 5: Actual v/s predicted Wind Speed for HDNN model

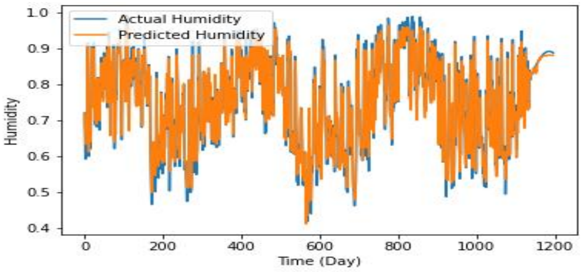

Figure 6: Actual v/s predicted Humidity for HDNN model

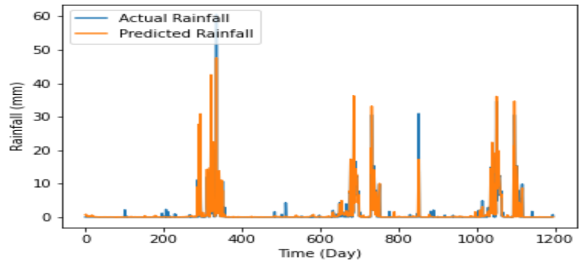

Figure 7: Actual v/s predicted Rainfall for HDNN model

The present model will generate 1197 samples in the testing data, matching the number of samples in the predicted data. The vast sample size makes it challenging to visualize all of these predictions in a single go, thus the prediction results for all four parameters for the 1197 days are displayed graphically. Figure (4 to 7), displays the prediction result of temperature, wind speed, humidity and rainfall respectively where the blue line represents the ground truth (Actual) temperature value for the particular day while orange line represents the prediction results for the corresponding day.

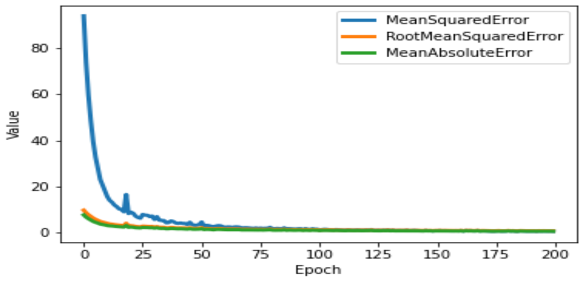

Figure 8: Performance metrics at various epoch for Temperature

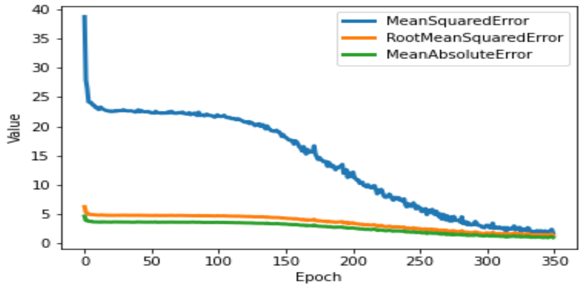

Figure 9: Performance metrics at various epoch for Wind Speed

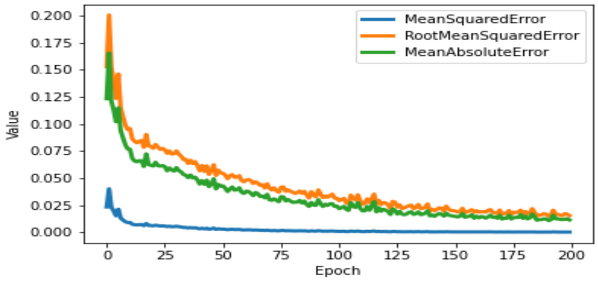

Figure 10: Performance metrics at various epoch for Humidity

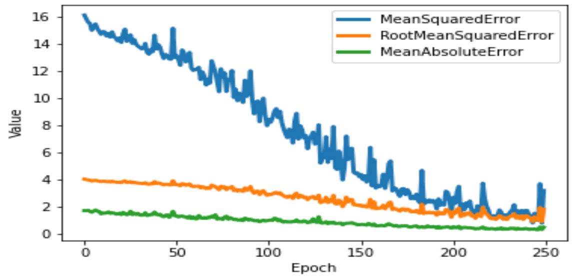

Figure 11: Performance metrics at various epoch for Rainfall

The graphs above in Figure (8 to 11) gives the values of performance metrics on various epochs for temperature, wind speed, humidity and rainfall in which on x-axis epoch range is shown and the y-axis shows the values of metrics.The least MSE is found at 177 epochs for temperature, 345 epochs for wind speed, 179 epochs for humidity and 243 epochs for rainfall parameter. As mentioned in section 3.3 following are the evaluation results of ARIMA and CRNN using the similar dataset which is used in the proposed HDNN model for performance comparison

Table 2: Evaluation results for ARIMA model

Parameters	Evaluation Metrics		
	MSE	RMSE	MAE
Temperature	2.252	1.500	0.806
Wind Speed	4.964	2.228	1.700
Humidity	0.003	0.055	0.044
Rainfall	3.753	1.937	0.755

The evaluation findings are provided in Table 2, which compares the anticipated results of all four weather parameters with the ground truth (actual values). The historical weather dataset utilized in this study is real/actual numbers and is simple to comprehend. The MSE measure is primarily used to assess the suggested model. Other metrics like MAE and RMSE are also used while analyzing the data.

Table 3: Evaluation results for CRNN model

Parameter	Evaluation Metrics		
	MSE	RMSE	MAE
Temperature	1.368	1.169	0.890
Wind Speed	2.706	1.645	1.256
Humidity	0.001	0.042	0.034
Rainfall	1.369	1.170	0.469

The values of all these evaluation metrics for the CRNN model are given in Table 3 which shows the skills of the CRNN model for forecasting future weather data. As there are 1197 samples in testing data, the CRNN model will generate similar number of outputs as the predicted data.

Table 4: Evaluation results for LSTM model

Parameters	Evaluation Metrics		
	MSE	RMSE	MAE
Temperature	0.839	0.916	0.703
Wind Speed	1.051	1.025	0.688
Humidity	0.0009	0.0312	0.0243
Rainfall	0.977	0.988	0.407

The LSTM model's abilities to anticipate future meteorological data are displayed in Table 4, which provides the values of all these evaluation measures. Given that the testing data contains 1197 samples, the LSTM model will produce an output that is comparable to the anticipated data.

Table 5: Evaluation results for Bi-LSTM model

Parameters	Evaluation Metrics		
	MSE	RMSE	MAE
Temperature	1.355	1.164	0.876
Wind Speed	2.357	1.535	1.185
Humidity	0.0010	0.0321	0.0246
Rainfall	1.939	1.392	0.525

Table 5 presents evaluation results for Bi-LSTM, comparing the expected outcomes of all four meteorological conditions. Together with RMSE and MAE for data analysis, the MSE metric is mostly used to evaluate the proposed model.

VI. ANALYSIS AND DISCUSSIONS

In this section performance of CRNN, ARIMA along with LSTM and Bi-LSTM models is compared with the proposed HDNN and analyzed. The evaluation of these models is done for each of the four parameters using the multiple input single output regression type, as explained in Section 3.3. The performance measures that are employed are MSE, RMSE, and MAE, as explained in Section 3.4. Furthermore, the fact that the success of aforementioned trials indicates that the suggested DL model can be applied to weather forecasting.

Table (6 to 9) gives comparative analysis of performance of ARIMA, CRNN, LSTM, Bi-LSTM and Proposed HDNN models for temperature, wind speed, humidity and rainfall forecasting respectively on different metrics while Figure (12 to 15) shows graphical analysis of MSE, RMSE and MAE for the same four parameters respectively.

Table 6: Temperature forecasting

Models	MSE	RMSE	MAE
ARIMA	2.252	1.500	0.806
CRNN	1.368	1.169	0.890
LSTM	0.839	0.916	0.703

Bi-LSTM	1.355	1.164	0.876
HDNN	0.486	0.697	0.517

Table 7: Wind Speed forecasting

Models	MSE	RMSE	MAE
ARIMA	4.964	2.228	1.700
CRNN	2.706	1.645	1.256
LSTM	1.051	1.025	0.688
Bi-LSTM	2.357	1.535	1.185
HDNN	0.856	0.925	0.677

Table 8:Humidity forecasting

Models	MSE	RMSE	MAE
ARIMA	0.0030	0.0551	0.0446
CRNN	0.0018	0.0428	0.0341
LSTM	0.0009	0.0312	0.0243
Bi-LSTM	0.0010	0.0321	0.0246
HDNN	0.0002	0.0155	0.0119

Table 9: Rainfall forecasting

Models	MSE	RMSE	MAE
ARIMA	3.753	1.937	0.755
CRNN	1.369	1.170	0.469
LSTM	0.977	0.988	0.407
Bi-LSTM	1.939	1.392	0.525
HDNN	0.763	0.873	0.209

In the present work DL models LSTM and Bidirectional LSTM are integrated for weather prediction. The two above mentioned models have been combined to greatly enhance the state of the art and it is clear from the findings that the HDNN model, when compared to the statistical model ARIMA, CRNN, LSTM and Bi-LSTM can provide more accurate forecasts. It has also been demonstrated that DL models are highly accurate at capturing complex or non-linear underlying physical process features. Additionally, literature review shows that the NWP models face a number of difficulties, such as the need for high computational power to execute numerous simultaneous non-linear equations, which takes a larger time to complete.

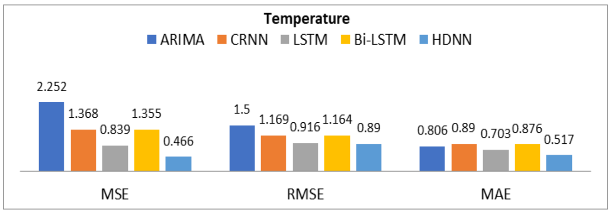

Figure 12: Evaluation metrics for Temperature

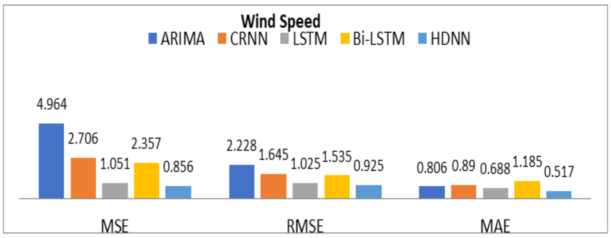

Figure 13: Evaluation metrics for Wind Speed

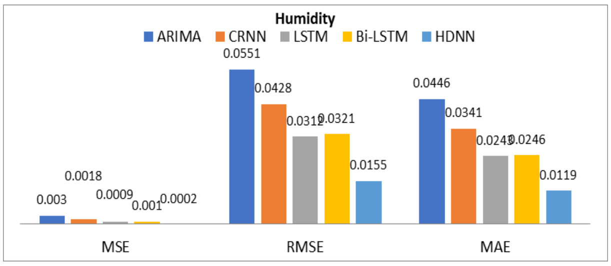

Figure 14: Evaluation metrics for Humidity

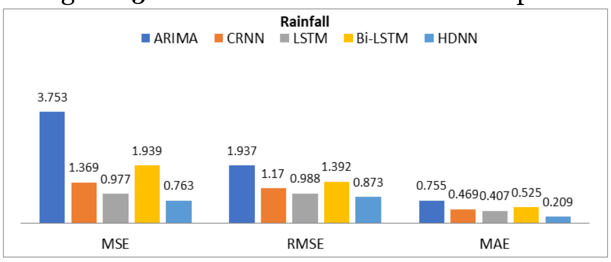

Figure 15: Evaluation metrics for Rainfall

The proposed approach can reduce the problems related to computational complexity, explained in Section 4. This supports the claim in literature review that adopting data-driven computer modelling techniques can lower the computational requirements of NWP methods. For historical weather data with MISO regression models, The proposed-HDNN model outperformed ARIMA, CRNN, LSTM and Bi-LSTM models considerably for each of the four parameters respectively as shown in Tables (6 to 9).

VI. CONCLUSION AND RECOMMENDATIONS

A. Conclusion

The purpose of this work is to develop and a HDNN model for analyzing historical weather data and demonstrates its superiority over other models as mentioned previously for all four forecasting parameters. The proposed model is designed to be easily implemented for a specific geographic region, allowing for fine-grained weather prediction on a standalone computer or a low cost and low power device. The other advantage of proposed model is its ability to address various challenges associated with existing weather forecasting models. It overcomes problems related to model comprehension, installation, and execution, making it easier to understand and use. The DNN approach of the model allows for portability and seamless integration with a Python environment, enabling the users to obtain useful results with minimal effort.

Compared to traditional NWP models, the suggested model is highly efficient as NWP models typically focus on regional forecasting and are not suitable for fine-grained geographical areas. Additionally, ML-based weather forecasting models often have limitations in predicting a limited number of weather conditions. In contrast, the proposed model can accurately predict up to four weather parameters, offering a more comprehensive forecasting capability. The experiment results validate the effectiveness of the DNN approach for weather prediction, suggesting that this model can be successfully applied in practice. Furthermore, the suggested model offers benefits over and above the existing models at regional and global level. It consumes fewer computational resources, making it more efficient in terms of computational requirements. Additionally, the model is easier to deploy and more portable, allowing for seamless integration into different environments. By incorporating location-specific data, the proposed model provides trustworthy and accurate predictions tailored to the specific geographic area of interest.

B. Limitations and Recommendations

Seven different surface weather parameters are used in this work. An increase in the quantity of inputs would certainly generate better outcomes. Nevertheless, this will rise the complexity of the model, necessitating the estimation of a huge number of parameters. Furthermore, the suggested model is trained using 70% of total weather data of 2792 days only. A DL network may be able to forecast more accurately if the training data sample size is increased. To enhance the prediction, further some more climatic variables could be added at various levels. In addition, although though it is less effective than MIMO, the MISO methodology is used in this study to anticipate weather conditions because it yields better MSE values. It follows that the MIMO approach will undoubtedly contribute to more precise prediction outcomes. Additionally, when paired with LSTM, Bi-LSTM produces improved accuracy and long-term prediction so even if time consumption is high, more accurate findings might be produced by the more effective usage of Bi-LSTM.

The prediction results could be better if the experiment uses controls with a range of alternatives or constant controls as variables. Again, it takes a lot of time to complete each of these trials. A potential solution would be to train the models using high-end hardware resources. On the other hand, the trained model is applied to prediction once these training experiments are conducted simultaneously. The process of making a prediction is not difficult or lengthy. Therefore, it is not recommended to invest excessively on mechanism that is highly specialized for the process of training.

Another significant limitation is the practical difficulties in determining suitable criteria for forecasting. It is not feasible to categorize in a way that would allow ML models to identify a subset of the various meteorological parameters. The effect of altering or removing a subset of input parameters on the final model output is almost impossible to compute. Another disadvantage of DL algorithms is that, larger dataset is preferred to train models for precise forecast and the training becomes difficult with the expanded size of the training dataset.

REFERENCES

[1] Brock, F. V., & Richardson, S. J. Meteorological measurement systems. Oxford University Press, USA. 2001

[2] Gao, S., & Chiu, L. S. Development of statistical typhoon intensity prediction: Application to satellite-observed surface evaporation and rain rate (STIPER). Weather and forecasting, 2012, 27(1), 240-250.

[3] Emeis, S., & Knoche, H. R. Applications in meteorology. Developments in soil-science, 2009, 33, 603-622.

[4] Parker, W. S. Predicting weather and climate: Uncertainty, ensembles and probability. Studies in history and philosophy of science part B: Studies in History and Philosophy of Modern Physics, 2010, 41(3), 263-272.

[5] Carey, C., & Alexander, M. A. Climate change and amphibian declines: is there a link?. Diversity and distributions, 2003, 9(2), 111-121.

[6] Castree, N., Adams, W. M., Barry, J., Brockington, D., Büscher, B., Corbera, E. & Wynne, B. Changing the intellectual climate. 2014, 763–768.

[7] Bellard, C., Bertelsmeier, C., Leadley, P., Thuiller, W., & Courchamp, F. Impacts of climate change on the future of biodiversity. Ecology letters, 2012, 15(4), 365-377.

[8] Pemberton, J. C., & Greenwald, L. G. On the need for dynamic scheduling of imaging satellites. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 2002, 34(1), 165-171.

[9] Naveen, L., & Mohan, H. S. Atmospheric weather prediction using various machine learning techniques: a survey. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC),2019, (pp. 422-428). IEEE.

[10] Scher, S., & Messori, G. Predicting weather forecast uncertainty with machine learning. Quarterly Journal of the Royal Meteorological Society, 2018, 144(717), 2830-2841.

[11] Lai, L. L., Braun, H., Zhang, Q. P., Wu, Q., Ma, Y. N., Sun, W. C., & Yang, L. Intelligent weather forecast. In Proceedings of International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826),2004, (Vol. 7, pp. 4216-4221).

[12] Liu, H., Chen, C., Tian, H. Q., & Li, Y. F. A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks. Renewable energy,2012, 48, 545-556.

[13] Liu, H., Tian, H. Q., & Li, Y. F. Comparison of two new ARIMA-ANN and ARIMA- Kalman hybrid methods for wind speed prediction. Applied Energy, 2012, 98, 415-424.

[14] Hewage, P., Trovati, M., Pereira, E., & Behera, A. Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications, 2021, 24(1), 343-366.

[15] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. Temporal convolutional networks for action segmentation and detection. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, (pp. 156-165).

[16] Husaini, N. A., Ghazali, R., Mohd Nawi, N., & Ismail, L. H. Jordan pi-sigma neural network for temperature prediction. In Ubiquitous Computing and Multimedia Applications: Second International Conference, UCMA Daejeon, Korea, April 13-15, 2011. Proceedings, Part II 2 (pp. 547-558).

[17] Kumar, A., Luo, J., & Bennett, G. F. Statistical evaluation of lower flammability distance (LFD) using four hazardous release models. Process Safety Progress, 1993, 12(1), 1-11.

[18] Hu, J., Wang, J., & Zeng, G. A hybrid forecasting approach applied to wind speed time series. Renewable Energy, 2013, 60, 185-194.

[19] Zaytar, M. A., & El Amrani, C. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. International Journal of Computer Applications, 2016, 143(11), 7-11.

[20] Ghaderi, A., Sanandaji, B. M., & Ghaderi, F. Deep forecast: Deep learning-based spatio-temporal forecasting., 2017, arXiv preprint arXiv:1707.08110.

[21] Chavan, G., & Momin, B. An integrated approach for weather forecasting over Internet of Things: A brief review. IEEE International conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC),2017, (pp. 83-88).

[22] Salman, A. G., Heryadi, Y., Abdurahman, E., & Suparta, W. Weather forecasting using merged long short-term memory model. Bulletin of Electrical Engineering and Informatics, 2018, 7(3), 377-385.

[23] Fente, D. N., & Singh, D. K. Weather forecasting using artificial neural network. IEEE Second international conference on inventive communication and computational technologies (ICICCT),2018, (pp. 1757-1761).

[24] Zhang, Z., & Dong, Y. Temperature forecasting via convolutional recurrent neural networks based on time-series data. Complexity, 2020, 1-8.

[25] Al Sadeque, Z., & Bui, F. M. A deep learning approach to predict weather data using cascaded LSTM network. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE),2020, (pp. 1-5). IEEE.

[26] Cifuentes, J., Marulanda, G., Bello, A., & Reneses, J. Air temperature forecasting using machine learning techniques: a review. Energies, 2020, 13(16), 4215.

[27] Thi Kieu Tran, T., Lee, T., Shin, J. Y., Kim, J. S., & Kamruzzaman, M., Deep learning based maximum Temperature forecasting assisted with meta-learning for hyperparameter optimization. Atmosphere, 2020, 11(5), 487.

[28] Liu, J. N., Hu, Y., You, J. J., & Chan, P. W. Deep neural network based feature representation for weather forecasting. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing, 2014.

[29] Molteni, F., Buizza, R., Palmer, T. N., & Petroliagis, T. The ECMWF ensemble prediction system: Methodology and validation. Quarterly journal of the royal meteorological society, 1996, 122(529), 73-119.

[30] Toth, Z., & Kalnay, E. Ensemble forecasting at NCEP and the breeding method. Monthly Weather Review, 1997, 125(12), 3297-3319.

[31] Pinson, P., & Madsen, H. Ensemble based probabilistic forecasting at Horns Rev. Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology, 2009, 12(2), 137-155.

[32] Niu, X., & Wang, J. A combined model based on data preprocessing strategy & multi objective optimization algorithm for short-term wind speed forecasting. Applied Energy, 2019, 241, 519-539.

[33] Sharma, U., & Sharma, C., Deep Learning Based Prediction Of Weather Using Hybrid_stacked Bi-Long Short Term Memory. IEEE 12th International Conference on Cloud Computing, Data Science & Engineering, 2022 , (pp. 422-427).

[34] Kumar, U., & Sharma, N. (2024). A Scientific Analysis of Weather Forecasting Techniques and its Classification. Journal of Management World, 2024(4), 917-923.

[35] Zhong, X., Chen, L., Li, H., Liu, J., Fan, X., Feng, J., ... & Lu, B. FuXi-ENS: A machine learning model for medium-range ensemble weather forecasting. arXiv preprint arXiv:2405.05925, 2024

[36] Liu, Y., Yuan, X., Jiao, Y., Ji, P., Li, C., & An, X. Ensemble Forecasts of Extreme Flood Events with Weather Forecasts, Land Surface Modeling and Deep Learning. Water, 2024, 16(7), 990.

[37] Mlakar, P., Merše, J., & Faganeli Pucer, J. Ensemble weather forecast post-processing with a flexible probabilistic neural network approach. Quarterly Journal of the Royal Meteorological Society, 2024, 150(764), 4156-4177.

[38] Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T. R., El-Kadi, A., Masters, D. & Willson, M. Probabilistic weather forecasting with machine learning. Nature, 2025, 637(8044), 84-90.

[39] McDaniel, P., Papernot, N., & Celik, Z. B., Machine learning in adversarial settings. IEEE Security & Privacy, 2016, 14(3), 68-72.

[40] Graves, A., & Graves, A. ,Supervised sequence labelling Springer Berlin Heidelberg, 2012, (pp. 5-13).

[41] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J., LSTM:A search space odyssey. IEEE transactions on neural networks and learning systems, 2016, 28(10), 2222-2232.

[42] Gers, F. A., Schmidhuber, J., & Cummins, F., Learning to forget: Continual prediction with LSTM. Neural computation, 2000, 12(10), 2451-2471.

[43] Zeiler, M. D., Adadelta: an adaptive learning rate method. arXiv preprint,2012, arXiv:1212.5701.

[44] Chin, W. S., Zhuang, Y., Juan, Y. C., & Lin, C. J., A learning-rate schedule for stochastic gradient methods to matrix factorization. In Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part I 19, 2015, (pp. 442-455). Springer International Publishing.