Received: 4 September 2022 | Revised: 27 October 2022 | Accepted: 4 February 2023

DOI: 10.1002/for.2966

RESEARCH ARTICLE

WILEY

Deep learning model for temperature prediction: A case study in New Delhi

Virendra Kumar Shrivastava[1] | Aastik Shrivastava[2] | Nonita Sharma[3] | Sachi Nandan Mohanty[4] | Chinmaya Ranjan Pattanaik[5]

[1]Department of Computer Science and Engineering Alliance College of Engineering and Design, Alliance University, Bangalore, India

[2]Department of Electronics and Telecommunication, International Institute of Information Technology, Bhubaneswar, Odisha, India

[3]Department of Information Technology, Indira Gandhi Delhi Technological University, New Delhi, India

[4]School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, Andhra Pradesh, India

[5]Department of Computer Science and Engineering, Ajay Binay Institute of Technology Cuttack, Odisha, India

Correspondence
Sachi Nandan Mohanty, School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India.
Email: sachinandan09@gmail.com

Abstract

This study is based on temperature prediction in the capital of India (New Delhi). We have adopted different ML models such as (MPR and DNN) which are designed and implemented for temperature prediction. The MPR models are varied on the degree of the polynomial, whereas the DNN models differ in the number of input parameters. DNNM-1 takes date, time, and temperature as input, and DNNM-2 receives date, time, temperature, pressure, humidity, and dew point as input parameters, whereas DNNM-3, is a complex model that takes date, time, temperature, pressure, humidity, dew point, and 32 weather conditions as input. To evaluate the accuracy of the predictions, a comparison of the predicted temperature and the actual recorded temperature is done, and the performance and accuracy of the models are examined. The MPR models work well in case of fewer input features, but as the number of input features grows, the DNN model outperforms the MPR models. The DNN model (DNNM-3) outperformed the other models with better accuracy as compared to past evidence.

KEYWORDS

artificial neural network (ANN), deep neural network (DNN), multivariate polynomial regression (MPR), temperature prediction

1 | INTRODUCTION

India experiences scorching heat waves in summer (May–June) every year in the northern part of the country whereas cold waves during winters (December–February) in the hilly areas like Ladakh, Jammu and Kashmir, Himachal Pradesh, and Uttarakhand. Similarly, coastal and riverbank areas suffer from floods during the monsoon season (June–August) every year, and millions of people's lives are affected by the heavy floods in Bihar, West Bengal, Assam, Uttar Pradesh, and so on. The country suffers a heavy loss of lives and crop ruins, which create a massive dent in the economy of the country. New Delhi, the heartland and the national capital of the country, also experiences scorching heat waves during summers, and the temperature is increasing every year (Biswas et al., 2014). The meteorological department edicts weather conditions using traditional and complex methods. It requires high-performance computing machines and expertise in multiple disciplines. The prediction of atmospheric conditions is crucial in different areas, that is, rain prediction, drought prediction, weather condition prediction in the aviation sector, agriculture, and tourism activities. The requirement of high-power computing resources and complexity makes traditional weather prediction approaches a time-consuming, complex, and costly process. On the other hand, machine learning (ML)-based models are fast, reliable, and cheaper methods to forecast weather (Biau, 2012; Biswas et al., 2014; Dolara et al., 2017; Gadekallu et al., 2019; Raed et al., 2010; Sahai et al., 2000; Shukla & Mooley, 1987; Voyant et al., 2012). Accurate prediction of weather is a tough job because of a nonlinear relationship between input features and the output atmospheric conditions (Hemalatha et al., 2021; Yalavarthi & Shashi, 2009). Supervised ML techniques are popular approaches for prediction or estimation problems. Regression is one of the most popular statistical methods to establish the relationship between input and output variables (Joaquin et al., 2007; Kapoor et al., 2014; Yeturu, 2019; Zhang et al., 2018). It is used to predict the relationship between input data and the predicted output. In ML, polynomial regression (Eva, 2012) is one of the types of regression that predicts relations between the independent variable (input data—X) and dependent variable (output—Y) by modeling *n*th degree polynomial (X). When X contains more than one variable or multiple independent factors, the class of polynomial is called multivariate polynomial regression (MPR). Weather forecasting can be done through different regression algorithms and neural networks (Marchuk, 2012; Zhou et al., 2019). The notion of artificial intelligence (AI) has drawn massive attention from scientists and computer science researchers' community as a powerful and efficient approach to substitute conventional methods for estimating and optimizing the dependent variables. Artificial neural network (ANN) model (Idicula & Mohanty, 2013; Raed et al., 2010) explored AI techniques in computing the rendition and superiority by using appropriate and admissible input parameters.

Regression is used to predict continuous dependent data with the help of independent data. Regression is the level of X (independent), which determines the level of Y (dependent). This is called functional dependency. A basic linear regression model looks like the following:

$$y = a_0 + a_1 x. \tag{1}$$

There are many other types of regression methods like multiple linear regression, polynomial regression, and decision tree regression. As the nonlinearity in the data increases, it becomes difficult for simple regression methods to find functional dependencies between dependent and independent variables. Multiple linear regression is suitable if several factors impact the response variable. In the case of complex problems, only finding linear relationships is not sufficient. Hence, in this paper, researchers have used MPR to forecast temperature. Polynomial regression is about increasing the degree of the relationship equation between the dependent and independent variables.

The neural network can address nonlinear relations (Jain et al., 1996). The basic building block of a neural network is a neuron. The concept of neural networks has been drawn from the neurons in the human brain. It has three types of layers of neurons: namely, the input layer, the hidden layers, and the output layer. In a neural network, each neuron is connected to other neurons to pass information. Each neuron in the hidden layer or intermediate layer is a mathematical function known as the activation function. There are several types of ANN, convolutional neural network (CNN) or deep neural network (DNN), recurrent neural network (RNN), and modular neural networks (MNNs) to name a few. A DNN or multilayer layer perceptron is based on ANN (Haleem et al., 2022; Shrivastava et al., 2021; Singhal et al., 2022; Wongchai et al., 2022). DNN is used for tasks like stock prediction, clustering problems, and weather prediction. A DNN is an extension of neural networks, which uses multiple hidden layers to find complex and high-level features from nonlinear data. So far, many researchers have contributed to weather condition prediction using mathematical models, statistical models, and ML models. Some research publications are focused on the prediction of the average temperature of the day and others considered the prediction of maximum and minimum temperature. Abrahamsen et al. (2018) used 1-year dataset for training and predicted temperature in 1, 3, 6, and 12 h (about 1.5 years) window for the next 48 h (about 2 days).

The proposed research work presents the design and implementation of ML-based models, namely, multivariate polynomial regression models (MPRMs) and deep neural network models (DNNM) to predict New Delhi temperature using the past 6 years' time-series data to predict the next year temperature in 6-h intervals. So far, no similar research work found to predict New Delhi temperature. The objective of the current work is to predict New Delhi temperature for the next year in the 6 h interval using the last 6 years' time series dataset with various input features, namely, date and time, temperature, atmospheric pressure, humidity, dew point, and 32 weather conditions like fog, heavy fog, and Drizzle by investigating ML techniques, namely, MPR and DNN. A comparison has also been made between the prediction result of the state-of-art DNNM and multivariate regression model and analyzed the results. To measure the performance and accuracy of the model, three different errors, namely, mean squared error (MSE), mean absolute error (MAE), and *R*-squared error (R^2), are calculated. The results of the proposed work delineate that the MPR works well in case of fewer input features, but as the number of input features grows, the DNN model outperforms the MPR model.

This paper is organized as follows: Section 2 provides necessary background information and review of previous work in weather forecasting. Section 3 (ML algorithms used for weather forecasting) describes relevant theoretical details of ML techniques, namely, MPR and DNN. In Section 4, the methodology is discussed. Detailed elaboration on the three steps of the methodology is presented. Further, Section 5 presents a detailed discussion and comparison of the results obtained by the different MPR and DNN models. It also critically analyzes and draws a comparison. In Section 6, the conclusion and future research work is discussed.

2 | PREVIOUS WORK

In recent years, researchers have shown great interest in the usage of ML models for weather prediction (Bochenek & Ustrnul, 2022; Hemalatha et al., 2021). A significant research contribution is made by some researchers using ML and DNNs. A few researchers used regression methods like functional regression, while some used support vector machines (SVMs). And recently, a combination of these predicting algorithms is also featured to predict the weather. Using the Google Scholar search engine, Bochenek and Ustrnul (2022) examined the 500 most relevant scholarly publications about ML techniques in the area of climate and numerical weather prediction that have been published after 2018. Authors observed that the most often studied meteorological conditions (wind, precipitation, temperature, pressure, and radiation) might be extracted using deep learning, random forest, ANNs, SVM, and Extreme Gradient Boosting (XGBoost).

Shad et al. (2022) explored seasonal autoregressive moving averages (SARIMA) and ANN with multilayer perceptron models for forecasting monthly relative humidity in Delhi, India, between 2017 and 2025. This study tried to use SARIMA and ANN with MLP models. Relative humidity trends between 2017 and 2025 have been predicted using average monthly relative humidity data for the years 2000 to 2016. According to the findings, the SARIMA model predicts relative humidity with an RMSE of 6.04 and a MAE of 4.56. The predicted relative humidity was given by the MLP model with a root mean squared error (RMSE) of 4.65 and MAE of 3.42. This research found that the ANN with MLP model proved more accurate in forecasting relative humidity than the SARIMA model.

Hemalatha et al. (2021) came up with a model for predicting the weather based on neural networks. Weather data from the Indian Metrological Department is used to test how well the proposed model works (IMD). Different metrics are used to test how well the model works. Madan et al. (2018) used a SVM and progressive statistical linear regression to forecast the weather for the next 5 days. Researchers also showed that better results can be obtained by using big data.

Yeturu (2019) focused on weather analysis using regression methods like linear regression, regression tree, MLP, and SVM in data mining. His research work considered factors such as average temperature, average pressure, and relative humidity. The performance of the algorithms has been determined based on MSE, RMSE, relative absolute error, and root-relative square error. In this work, authors predicted average temperature yearwise and month-wise, whereas the proposed work predicts temperature with 6-h interval, date wise for entire year based on 6-year time series data.

Denny et al. (2019) explored weather conditions to predict soccer matches using random forest, support vector, and K-nearest neighbors and presented that weather conditions can also be considerable factors to predict soccer matches.

Liu et al. (2019) used an ANN to predict rainfall. The research consists of the background of ANN and other neural network algorithms used for rainfall prediction in recent years. It has also been shown that neural networks can produce high accuracy and are efficient for prediction.

Abrahamsen et al. (2018) compared the prediction results for four different models based on ANN architecture that utilize weather data at the interval of 1, 3, 6, and 12 h using the 1-year dataset to predict the weather of 1 day before in the window of 48 h (about 2 days). However, the dataset used in this work is quite small.

Geetha and Nasira (2017) presented a time series model to predict the precipitation in coastal regions of India. The model was built on the Autoregressive Integrated Moving Average (ARIMA) model. The model is built upon a dataset that consists of temperature, rainfall, wind speed, visibility, and so forth for 5 years. The performance of the model has been calculated based on MSE, mean absolute deviation, RMSE, and mean absolute percentage error. However, they used ARIMA model, which is complex and predicted precipitation, not temperature.

Holmstrom et al. (2016) explored the previous 2 days data to predict the minimum and maximum temperature for the next 7 days. The authors used two regression models, namely, functional regression and linear regression, and compared their results. Researchers also concluded that linear regression could produce significantly better predictions if more data are fed.

Grover et al. (2015) investigated a hybrid approach for weather forecasting. Authors exhibited that a DNN can be enhanced with spatial interpolation and derived an effective learning procedure that enables large scale optimization of model parameters.

Bhatkande and Hubballi (2016) described that the decision tree algorithm for data mining can be used for pruning unnecessary data from datasets and used minimum and maximum temperatures for weather prediction.

Biswas et al. (2014) used multiple weather attributes forecasting methods for 1 day ahead prediction by using Nonlinear Autoregressive Networks with eXogenous (NARX) inputs neural networks. Researchers presented a case that can capture the dynamic and chaotic behavior of weather. The prediction results of case-based reasoning (CBR) and NARX neural networks have been compared, and the results proved the superiority of NARX method for forecasting multiple weather attributes.

Htike (2018) proposed the use of Focused Time-Delay Neural Networks (FTDNNs). The prediction for different time scales has been compared and presented the optimal neural network model parameters for each time scale. The dataset has been obtained from the Malaysian Meteorological department for 30 years. MSE has been used to evaluate the prediction results.

Wang, Lu, et al. (2020) discussed the importance of deep learning and recent advances in deep learning. The researchers emphasized how deep learning has revolutionized the future of AI in dealing with complex problems that existed for many years. Deep learning models are suitable for predicting linear and non-linear data. In this paper, researchers highlighted various application areas of deep learning and their variants.

Eva (2012) concentrated on diverse types of regression and explained the types of polynomial regression modeling techniques. This paper has elaborated on the ways to evaluate the accuracy of regression models and discussed the least squared error method, polynomial regression model, error calculation methods, and impact on model performance. This paper also emphasized its application areas of polynomial regression.

Wang, Zaho, and Pourpanah (2020) discussed multi-output least squares Support Vector Regression (MSVR) to forecast the wind speed and direction. MSVR can understand more complex patterns in data than single output SVR. Hence, it has been shown that the MSVR model based on bat algorithms is more effective and better than other algorithms.

Some papers used the data of geographical locations near the area where prediction is to be done. A lot of research is done using neural networks for temperature and rainfall prediction. In recent years, big data is being used with regression and other ML models.

During the literature survey, researchers have not come across any similar work to predict temperature for New Delhi using MPR and DNN models. To the best of authors' knowledge, the MPR and DNN models with the same parameters have not been studied on the New Delhi weather time series dataset so far. This paper used past 6-year data as input for training the models and predicting the next year temperature. The predicted results are compared with actual recorded data.

3 | METHODS

3.1 | Multivariate polynomial regression

To find simpler patterns and relations between dependent and independent variables, linear regression is used. When more than one factor or independent variables (x) are producing a change in the response variables, multiple linear regression is used instead of simple linear regression (Kostas et al., 2018). But, for determining non-linear and complex relations between the variables, often, linear regression methods are not enough, and a higher order relationship between x and y needs to be established, and therefore, the regression becomes polynomial and produces a better fitting model. An algorithm with multiple independent variables and higher order relationships forms the MPR. The formulation of the polynomial regression is as follows:

An nth order univariate polynomial regression can be expressed as

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n. \tag{2}$$

Multiple Linear regression can be stated as

$$y(x_1, x_2, \ldots, x_m) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \ldots + a_m x_m + \epsilon \tag{3}$$

ϵ is the residual term.

Furthermore, MPR can be formulated as

$$y(x_1, x_2, \ldots, x_m) = a_0 + \sum_{l1=1}^{m} a_{l1} x_{l1} + \sum_{l1=1}^{m} \sum_{l2=l1}^{m} a_{l1\,l2} x_{l1} x_{l2} + \ldots \tag{4}$$

ϵ is the residual term.

Equation (4) is an nth-order equation with m variables. The degree of MPR can be changed and varies from model to model. But as the degree of the polynomial is increased, the model adapts the relations between

x and y better, but after some degree, it becomes overfit. To determine an optimal solution, it is essential to employ bias and variance. Bias refers to the difference between the model's prediction and the true values, while variance is the variations in the model's prediction for a data value from the true value. Thus, a good model should maintain a tradeoff between the two. It is a recommended approach to find the best possible order for the model through cross-validation.

3.2 | Deep neural network

ANN is used to find nonlinear patterns in data. The actual strength of ANN comes from the hidden layers, which are present in between the input and output layers (Jain et al., 1996, Shrivastava et al., 2022). ANN is a feed-forward network. A DNN is an ANN that consists of more than one hidden layer and is also called a multi-layer perceptron (MLP). The multiple hidden layers carry out feature transformation and extraction. The first layer of the DNN processes raw data (input) and passes it to the second layer called the hidden layer. The second layer processes the information further according to the activation function and passes it to the next layer Anwar et al. (2017). This process is adopted by all the layers in the DNN until the desired result (output) is achieved. Each neuron has an activation function, and activation functions are applied to the data to standardize the output coming out of the neuron. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship (Hossain et al., 2015; Lee et al., 2020). The final output of a DNN consists of hidden layers p, containing m neurons and n input neurons, and b bias is expressed in Equation (5), and the architecture of the DNN can be visualized in Figure 1.

$$y = \sum .... \sum ....(p\ times)..\ ...\sum_{k=0}^{m}\left(w_k f_k\left(\sum_{j=0}^{m}\left(w_j f_j\left(\sum_{i=0}^{n} x_i w_i\right)\right)\right)\right) + b \tag{5}$$

Each neuron in a hidden layer contains an activation function (fa[]). The output of a neuron presents in a hidden layer connected to n input layer neurons can be expressed as Equation (6) (Abrahamsen et al., 2018).

$$y = fa\left(\sum_{i=0}^{n} x_i w_i + b\right), \tag{6}$$

where x_i is i^{th} independent variable and w_i is its corresponding weight; b represents the bias term. A hidden neuron demonstrating the output of an activation function is modeled in Figure 2.

The popular activation functions, namely, sigmoid or logistic, hyperbolic tangent, or tanh and Rectified Linear Units (ReLU), are used in DNN. The functions are depicted in Figure 3.

To achieve superior results, it is imperative to apply optimization algorithms. Many optimization algorithms have been developed. The most popular optimizers are Stochastic Gradient Descent, adaptive moment estimation (Adam), Adamax, Adagrad, among others (Khan et al., 2013, 2015; Zhang et al., 2020). While using optimizers, there is a frequent problem that arises; instead of finding global minima, the model may fall into the trap of local minima, which makes an adverse impact on the accuracy. It is exceedingly difficult to find a perfect model, but research is going in to make the algorithms better. Therefore, there are many algorithms available to improve model performance, but it is always recommended to test the model for different parameters and optimizers.

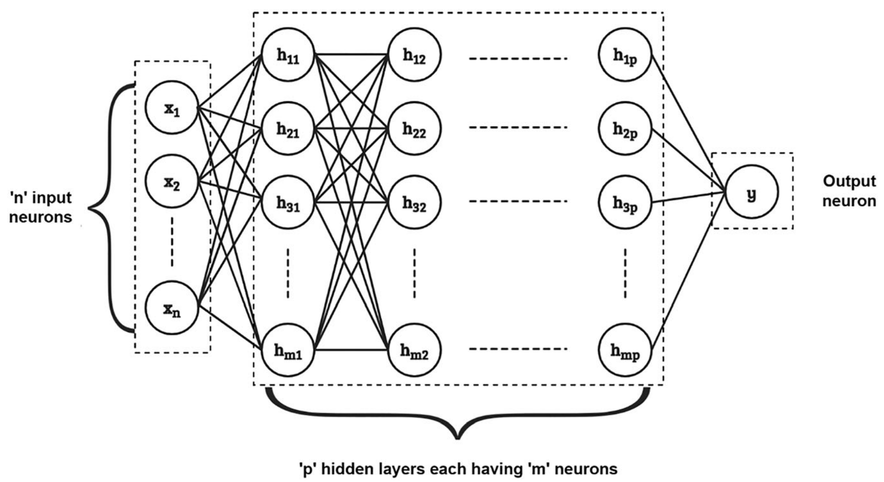

FIGURE 1 DNN or MLP architecture

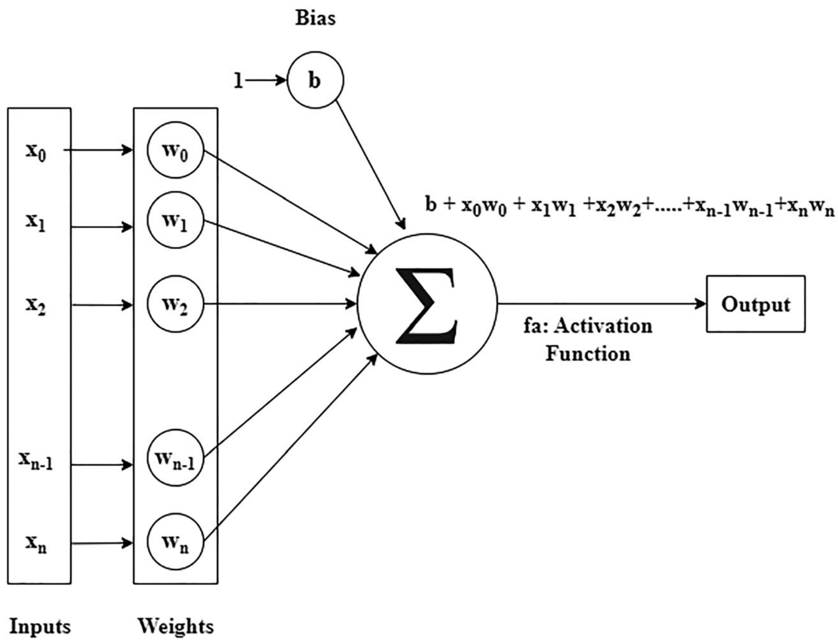

FIGURE 2 A hidden neuron demonstrating the output of an activation function

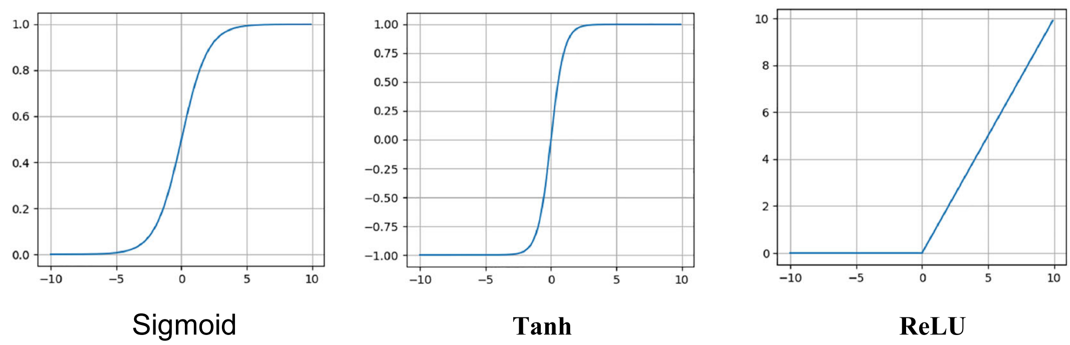

FIGURE 3 The graphical representation of the above activation functions

While increasing the model efficiency, one of the biggest challenges is when the model completely adapts the training set and fits it too well but fails to generalize the prediction in new data samples. In such a case, the training set accuracy is much better than the test set accuracy, and this scenario is called overfitting. There are many methods to prevent overfittings such as L1 and L2 regularization. In other words, these parameters limit the performance of models by adding a penalty parameter omega (Ω) in the objective function (J). The objective function calculates the quality of any solution by taking model parameters as arguments.

$$\tilde{J}(\theta;X,y) = J(\theta;X,y) + \alpha\Omega(\theta) \tag{7}$$

Equation (7) is the objective function (J). The regularization term in L1 regularization is defined in Equation (8).

$$\Omega(\theta) = \|\omega\|_1 = \sum\nolimits_i^1 |\omega_i| \tag{8}$$

Hence, the corresponding objective function with L1 regularization is presented in Equation (9).

$$\tilde{J}(\theta;X,y) = \alpha\|\omega\|_1 + J(\theta;X,y) \tag{9}$$

In L2 parameter regularization, the regularization term is defined in Equation (10).

$$\Omega(\theta) = \frac{1}{2}\|\omega\|_2^2, \tag{10}$$

where

$$\|\omega\|_2^2 = \omega_1^2 + \omega_2^2 + ... + \omega_n^2. \tag{11}$$

And hence, the corresponding objective function with L2 regularization is expressed in Equation (12).

$$\tilde{J}(\theta;X,y) = \frac{\alpha}{2}\omega^T\omega + J(\theta;X,y) \tag{12}$$

Equations (7) delineates that the value of "α" should be controlled carefully; otherwise, for high values of "α," the model can become underfit. Another important method is the early stopping of training in which algorithm is terminated before it gets completely executed over complete data so that error does not increase. And dropout is also an important technique in which some neurons are dropped after each epoch; hence, some specific features are removed from the model and the model will not overfit.

4 | METHODOLOGY

The proposed work makes use of a three-stage empirical procedure as its modeling technique. The first stage of the proposed methodology is obtaining dataset and preprocessing relevant data. In the next stage, different variants of MPR and DNN are designed, and input data are supplied to different variants of MPR and DNN models. The New Delhi time series dataset is used as structured feature inputs to train these models. In the final step, the hyper-parameters are optimized for best prediction performance and tested against a test set that is not included in the training set. MPR and DNN models follow the workflow presented in Figure 4.

In this work, five different models (two based on MPR and three based on DNN) have been created. The basic methodology for all the models are the same. The DNN model differs in the number of input parameters used for training, and the MPR models vary in the degree of relationship equation of the number of input parameters and response variable. The prediction models are implemented in Python 3.8. Matplotlib, Keras, and Pandas libraries are utilized for handling and preprocessing the dataset and implementation of the MPR and DNN models, data visualization, and graph plotting.

The workflow of the methodology is divided broadly into three steps:

Set 1: Data collection and preprocessing.
Set 2: Designing, training, and testing the models.
Set 3: Tuning and model performance evaluation.

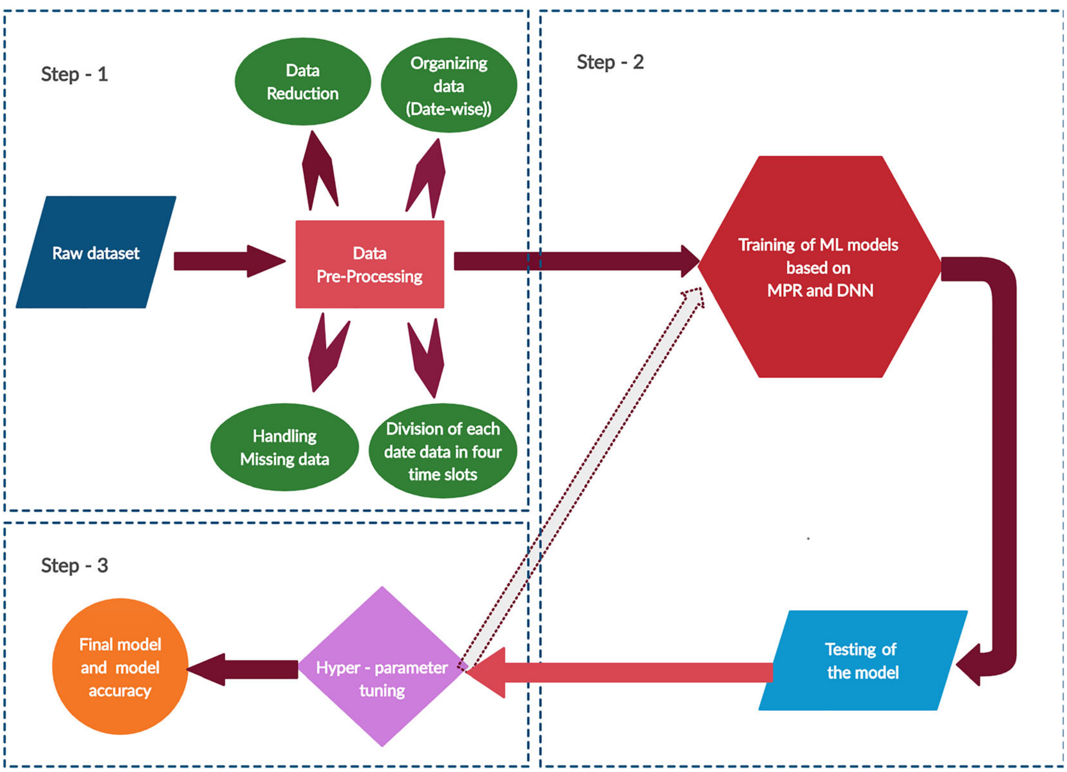

FIGURE 4 Workflow of proposed models

4.1 | Step 1: Data collection and preprocessing

4.1.1 | Raw dataset

The dataset used for this work is freely available on Kaggle and is owned by the weather underground (Kaggle.com, 2020). The dataset consists of temperature, humidity, pressure, dew point, and 32 atmospheric conditions of New Delhi from January 1, 1997, to December 31, 2016, on an hourly interval containing a total of 98,292 rows. The atmospheric conditions have 32 unique categories like fog, mist, and smoke. These conditions specify the atmospheric state for each record in the dataset.

4.1.2 | Data preprocessing

This phase of the methodology consists of four steps. First, we organized the data, date-wise and time-wise, and analyzed the nonlinear features of the dataset. Further, from the given dataset, temperature values between 2010 and 2016 have been selected. Each date of 2016 starting from January 1 to December 31 is placed in the dependent variable column and for each entry in that column of time-series data of 6 years (2010–2015) has been added containing temperature, humidity, pressure, dew point, and other atmospheric conditions as categorical features like fog, mist, smoke, thunder, and rain as independent variables columns.

Missing data in the dataset are handled by using the imputer class of preprocessing module of scikit-learn and is replaced by the mean value of all other entities of the respective field. Due to lack of a similar pattern in the time interval of a particular day, the data have been divided into four slots of 6 h, which is described in Table 1. Each slot has the mean value of all the data within that slot. Eighty percent of data have been used for the training, and the rest of 20% data are used to the test the model.

4.2 | Step 2: Designing, training, and testing the models

In this research work, two variants of MPR, namely, MPRM-1 and MPRM-2, and three variants of DNNs, namely, DNNM-1, DNNM-2, and DNNM-3, are designed. Eighty percent of data are used for training the models, and 20% of data are used for model testing. This is the phase where the dataset is passed to the algorithm and the algorithm leverages sophisticated ML models to train and predict the desired outcome (temperature). After training the models, performance (error) is calculated, and if the model's performance is not found satisfactory, the model is recalibrated and again tested until the result is found satisfactory.

TABLE 1 Time slot division

Slot number	Time slot
1	00:00–06:00
2	06:00–12:00
3	12:00–18:00
4	18:00–24:00

4.2.1 | Multivariate polynomial regression models (MPRM-1 and MPRM-2)

Two MPRM, namely, MPRM-1 and MPRM-2, are designed. Using MPR algorithm, the model is trained on time-series temperature data of the past 6 years, to predict next year's temperature. Atmospheric conditions like humidity and pressure are not considered. Because, as the number of variables increase in the MPR model, the number of terms also increases according to Equation (4). According to the equation, for two variables, a second-degree MPR function can be written as

$$y(x_1,x2) = a_0 + a_1x_1 + a_2x_2 + a_{11}x_1^2 + a_{22}x_2^2 + a_{12}x_1x_2. \tag{13}$$

Similarly for the variables, the second-order MPR function is

$$y(x_1,x_2,x_3) = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_{11}x_1^2 + a_{22}x_2^2 + a_{33}x_3^2 + a_{12}x_1x_2 + a_{23}x_2x_3 + a_{13}x_1x_3. \tag{14}$$

It is observed from the above equation, in a second-degree MPR, by increasing one more variable in the equation, four added terms are introduced in the equation, and as the degree and number of variables are increased, more added terms will be generated. It has been observed that for six variables there are more than 50 terms for a third-degree equation. Hence, the computational power becomes much high for higher degree models with a greater number of variables, and these models are also prone to overfitting. Thus, only temperature is taken as input feature for MPR models. It is noticed that the MPR models of degree 2 or 3 produced satisfactory predictions in comparison with higher degree (degree >3) MPR model variants. The MPRM variants and degree of polynomial have been described in Table 2. The MPRM variants are trained and tested (for degree = 2 and degree = 3) on time-series temperature

TABLE 2 Categories of MPRM

MPRM variants	Degree of the polynomial
MPRM-1	Degree = 2
MPRM-2	Degree = 3

data. The results obtained from the MPRM-1 and MPRM-2 are given in Table 3, which are based on the performance measures, namely, MSE, MAE, and R^2.

It is perceived from Table 3 that MPRM with degree = 2 (MPRM-1) performs better predictions than

TABLE 3 Performance measures, namely, MAE, MSE, and squared error of MPRM variants

	MAE (%)		MSE (%)		R^2 (%)	
Slot No.	MPRM-1	MPRM-2	MPRM-1	MPRM-2	MPRM-1	MPRM-2
1	1.4922	1.7254	3.4686	5.1199	0.9227	0.8859
2	2.0068	2.2238	6.2811	8.0393	0.8198	0.7694
3	2.2362	2.9150	7.4473	12.8129	0.8001	0.6562
4	1.5690	1.5335	3.7979	3.5197	0.9141	0.9222

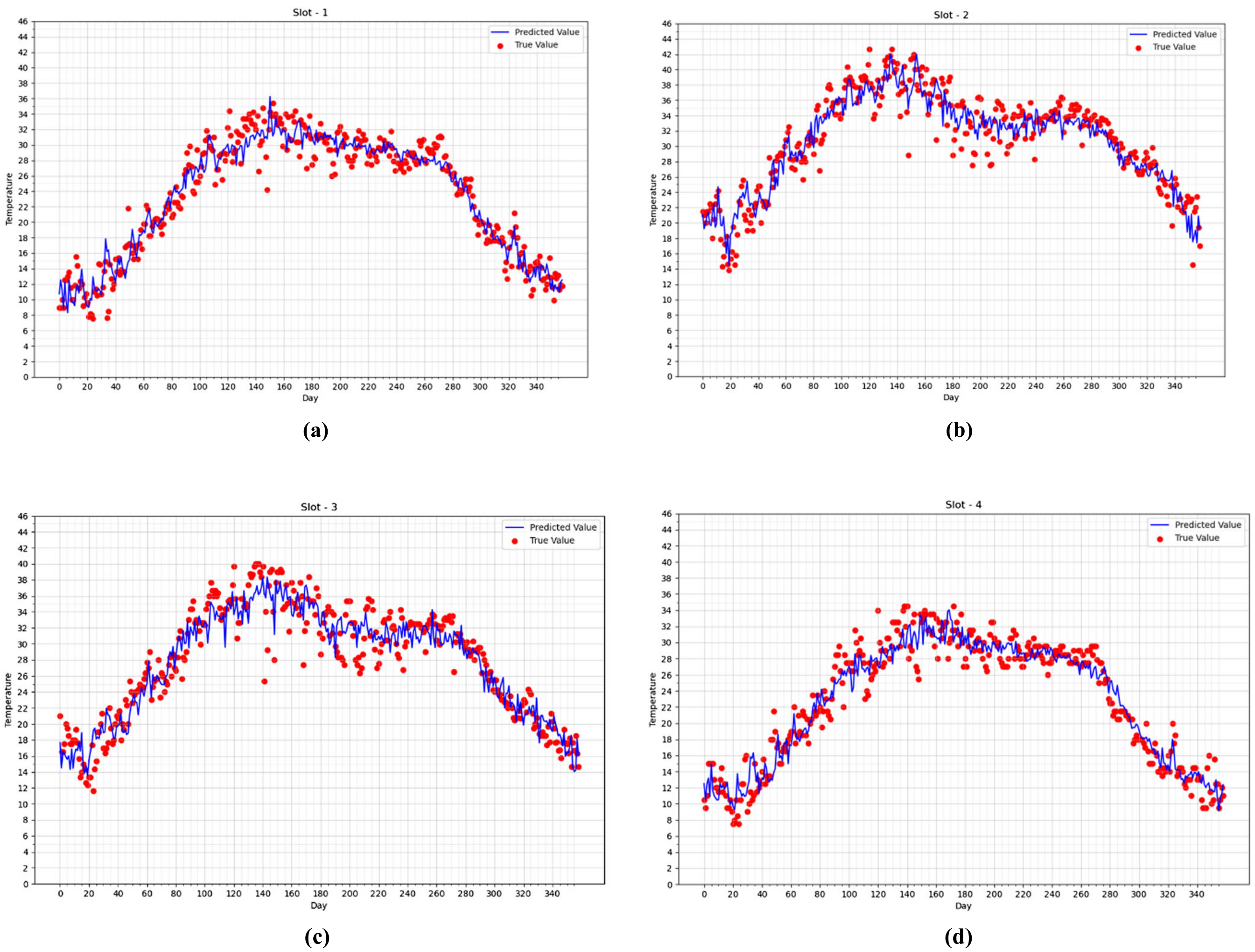

FIGURE 5 (a) Temperature prediction using the MPRM-1 for slot 1. (b) Temperature prediction using the MPRM-1 for slot 2. (c) Temperature prediction using the MPRM-1 for slot 3. (d) Temperature prediction using the MPRM-1 for slot 4

the MPRM with degree = 3 (MPRM-2). At higher degrees, the degree of randomness or noise in the prediction curve increases for the changes in the training data values and, hence, produced poor test set results. These results are better than the MPRM containing other weather parameters in addition to temperature data, which tend to fluctuate too much and generated even worse results. The temperature predictions of MPRM-1 have been shown in Figure 5a–d for slots 1–4, respectively. Table 3 is evident that the performance of MPRM-1 is better than MPRM-2. The y axis of Figure 5a–d denotes temperature in degrees Celsius, and the x axis denotes days of the year (2016) of which prediction is done.

4.2.2 | DNN model

Three variants of DNNs are designed and implemented. The model varies in the number of parameters taken as input features to the model. DNNM-1 is a simple DNN model in which the temperature of the last 6 years (2010–2015) in intervals of 6 h (four windows) is given as input. So six input parameters are given to the input layer. In DNNM-2, input features such as temperature, pressure, humidity, and due point of the last 6 years (2010–2015) are given to the input layer. That means a total of 24 input features are given to the input layer (refer to Table 4), whereas in DNNM-3, input features such as temperature, pressure, humidity, due point, and 32 weather conditions of the last 6 years (2010–2015) are given to the input layer. In DNNM-3 we have adopted 210 input features (refer to Table 4). Table 4 shows the three models and parameters that have been taken into consideration. Figure 6 shows the DNN model architecture used for the three DNNM variants.

Total weather conditions given in the time series dataset are 32, but to avoid a dummy variable trap, one column is removed; hence, total weather condition input features are 31 in Table 4. Weather condition input features are depicted in Table 5. The DNN models are designed with five hidden layers, 256 neurons, and a batch size is 64. The rectified linear unit (ReLU), and the activation function are used, and the Adam optimizer is chosen to optimize the result of the DNN models. Weights were initialized to small values with mean = 0 and standard deviation = 0.025. The model also consisted of L1 and L2 parameter regularization with a value of 0.001 and 0.0001. At higher values, it becomes underfit,

TABLE 5 Weather conditions input features in time series dataset

Weather conditions
Partial fog, smoke, mist, patches of fog, light fog, heavy fog, light drizzle, light rain, haze, partly cloudy, thunderstorm, thunderstorms and rain, widespread dust, light thunderstorm, blowing sand, heavy thunderstorms and rain, overcast, rain, scattered clouds, mostly cloudy, thunderstorms with hail, light hail showers, light sandstorm, drizzle, clear, light rain showers, heavy rain, light freezing rain, fog, shallow fog, rain showers, light thunderstorms and rain

TABLE 4 Input features for DNN models

DNNM variants	Parameters	l	$N = l * 6$
DNNM-1	Temperature	1	6
DNNM-2	Temperature, pressure, humidity, dew point	4	24
DNNM-3	Temperature, pressure, humidity, dew point, and 32 weather conditions input feature	4 + 31 = 35	210

Note: Here, l is number of parameters considered. N is total number of input variables. $N = l *$ (number of past years' data). Number of past years considered = 6 (2010–2015).

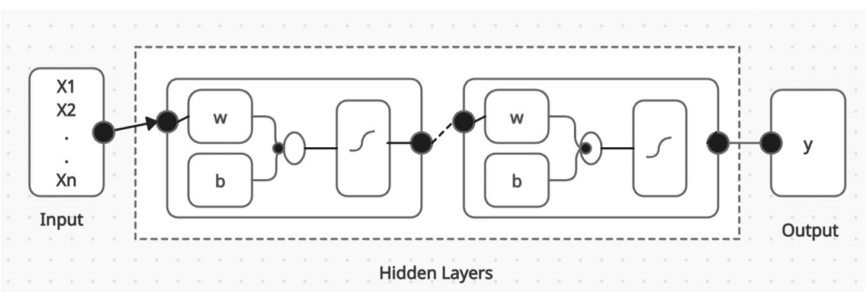

FIGURE 6 Architecture of DNNM variants

and at lower values, it tends to become overfit. The Adam optimizer was utilised with a low learning rate of 0.0001. Although a higher learning rate caused the model's loss function to start converging much faster, there are small but rapid oscillations in the test set loss that increased the test set loss, and the model failed to achieve its accurate global minima. But when a lower learning rate was applied, then not only the oscillations were reduced but the training and test set loss also came closer, and the model performed better. The model was trained on 300 epochs, beyond which there was no notable change observed in the model's performance, and after 300 epochs, the model slowly tended to overfit. When fewer neurons were used, the model obtained a little higher testing accuracy than the training accuracy, which happens due to overfitting, but this was not the case here.

TABLE 6 Performance measures, namely, MAE, MSE, and squared error of DNNM variants

	MAE (%)			MSE (%)			R^2 (%)		
SlotNo.	DNNM-1	DNNM-2	DNNM-3	DNNM-1	DNNM-2	DNNM-3	DNNM-1	DNNM-2	DNNM-3
1	1.4738	1.4612	0.9153	3.0517	3.0048	1.3024	0.9320	0.9330	0.9709
2	2.1104	1.8191	1.4689	6.8092	5.1473	3.3169	0.8047	0.8523	0.9048
3	2.2994	2.0497	1.2315	7.6001	6.5874	2.5318	0.7960	0.8232	0.9320
4	1.5807	1.5389	0.9652	3.8489	3.5119	1.5718	0.9130	0.9283	0.9644

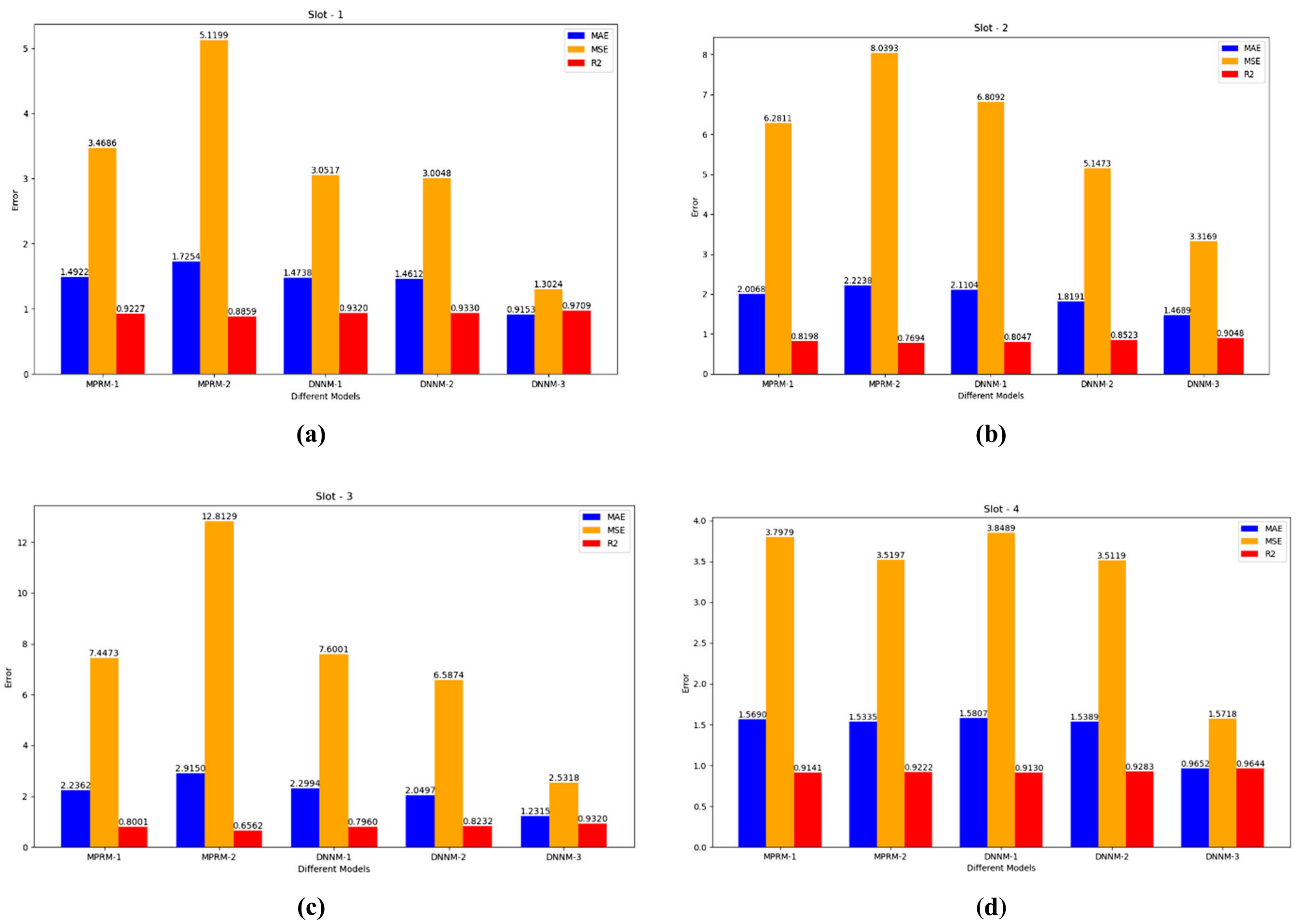

FIGURE 7 (a) Prediction result of DNNM-3 for time slot 1. (b) Prediction result of DNNM-3 for time slot 2. (c) Prediction result of DNNM-3 for time slot 3. (c) Prediction result of DNNM-3 for time slot 4

The model did this because of the complex training data. It could not understand the data completely.

The dataset consists of a variety of weather data for each date of the year from the summer to the winter season. There is a complex pattern in the training data. The result obtained from the DNNM-1, DNNM-2, and DNNM-3 have been given in Table 6 based on performance measures, namely, MSE, MAE, and R^2.

Table 6 data revealed that the performance and accuracy of DNNM-3 are much higher than DNNM-2 and DNNM-1. The error plots show (refer Figure 7a–d) that these atmospheric/weather conditions (refer Table 6) have significant impact on the weather conditions. When time-series data, namely, humidity, pressure, and dew point, were included with temperature, then the model performed a bit better, but the improvement was not significant. But DNNM-3 with the addition of time series weather conditions, the model performed to its best and produced impressive results for all the time slots (refer Table 1).

Figure 7a–d demonstrates the prediction curves for DNNM-3 (for four-time slots) as it has the best performance among of all DNN models:

5 | RESULTS AND DISCUSSIONS

In this research work, MPRM and DNN models are designed to predict temperature using time-series data. Two variants of MPRM are designed. MPRM models took

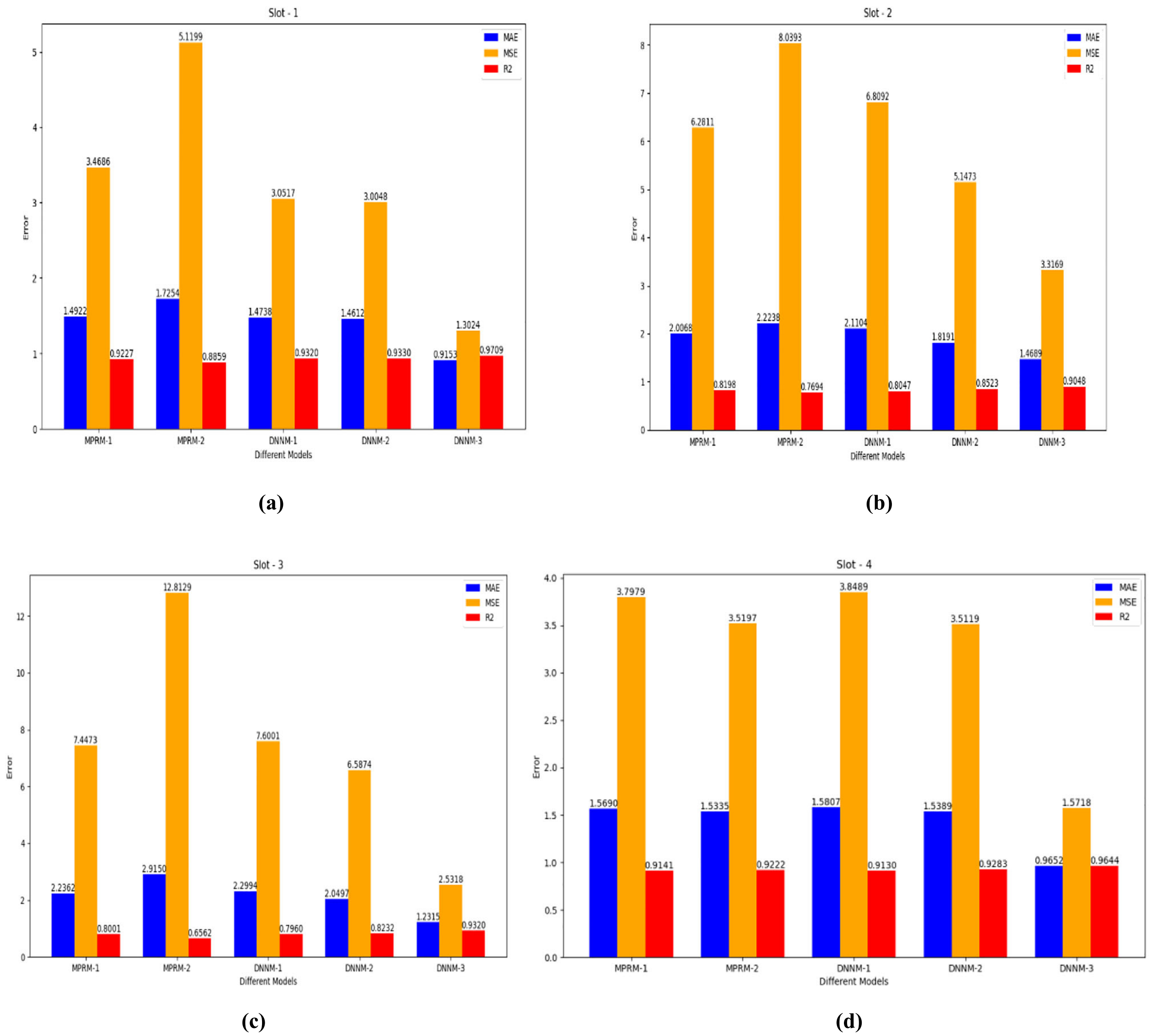

FIGURE 8 (a) MSE, MAE, and R^2 for slot 1. (b) MSE, MAE, and R^2 for slot 2. (c) MSE, MAE, and R^2 for slot 3. (d) MSE, MAE, and R^2 for slot 4

6 years temperature (2010–2015) as input parameters to predict temperature of 2016 in 6-h widow. Three variants of DNN are designed, which used time series dataset that consists of temperature, humidity, pressure, dew point, and 32 atmospheric/weather conditions like fog, mist, and smoke (refer Table 4 for 6 years; 2010–2015). Three variants of DNN model are trained and tested to predict the next year (i.e., 2016) temperature in a 6-h interval, and the prediction performance of the models is measured on MSE, MAE, and R^2 (predicted data vs. actual recorded data). The MSE, MAE, and R^2 for all the five models slot-wise are depicted in Figure 8a–d for slots 1–4, respectively.

Figure 8a–d delineates that when the only temperature is taken into consideration to training the models (MPRM-1, MPRM-2, and DNNM-1), the MPRM performs slightly better than the DNNM-1. Further, when additional weather conditions are taken into consideration, then the performance and accuracy of the DNNM-2 improved slowly. When all the atmospheric/weather conditions such as categorical features are included to train the DNNM-3, then a significant improvement has been observed with the same model hyper-parameters, which proves that the atmospheric conditions have a notable impact on the weather of a place. And if atmospheric conditions were ignored while forecasting the weather, it caused adverse impact over predicted results.

The major drawback of MPRM is that, at a higher degree and with more variables, the model becomes computationally expensive, and improving the accuracy of the model tends to fluctuate too much subject to the variations in the data, which deteriorates the model's predictions. Even only time-series temperature data are used as input to train MPRM-1 and MPRM-2, the performance of the model and accuracy worsened when the degree is increased from 2 (MPRM-1) to 3 (MPRM-2), and it can be observed from Table 3 that MPRM-1 with degree = 2 performs better than MPRM-2 with degree = 3, which clarifies that degree of randomness or fluctuations in prediction increases in MPRM-2 and hence errors increased. On the contrary, deep neural model with a fewer number of parameters (DNNM-1) did not perform up to the mark. But, as the number of input features increased in the subsequent models (DNNM-2 and DNNM-3), the performance of the models increased, which easily outperformed MPR models. DNNM-3 produces the best results (96.4%) among all the five models without much change in hyper-parameters and computational power, which clearly shows that for large data and more independent variables or parameters, DNNM-3 performed better than the other ML models.

As mentioned in the literature survey, researchers did not find a similar study on the New Delhi temperature prediction. However, Shad et al. (2022) explored ANN with a multilayer perceptron model for forecasting monthly relative humidity in Delhi, India, between 2017 and 2025. The predicted relative humidity was given by the MLP model with a RMSE of 4.65 and a MAE of 3.42, whereas DNNM-1, DNNM-2, and DNN-3 models predicted temperature with MAE between 0.9652 and 2.2992 (refer Table 6), which is quite lesser than the Shad et al. (2022) ANN with MLP model.

6 | CONCLUSION AND FUTURE WORK

This research paper exhibits a comprehensive and comparative analysis of different models of MPR and DNN, trained and tested on New Delhi weather temporal dataset. This work carried out an empirical study on 6-year time-series weather dataset of New Delhi and designed, trained, and tested MPR and DNN models for temperature prediction and compared the results. There is no similar research work found to the best of the authors' knowledge related to New Delhi temperature prediction. There is some work available on the use of time-series data [11] for weather prediction, which is based on models like ARIMA and SARIMA. This paper presented three DNN models, trained 6-year time-series dataset to predict the temperature of the next year of New Delhi and showed which factors are important and deterministic for weather prediction. The results of DNN models are also compared with MPR models. In the case of a large number of input features, the comparison confirms superiority of DNN models over MPR models. The DNNM-3 produced the best temperature prediction results (96.4%) among all five models (refer Table 6).

This research delineates temperature prediction of New Delhi with time-series data of the past 6 years. The temperature of a city is also influenced by the environmental conditions of the neighboring cities and towns. It can also be observed that how the temperature of neighboring locations affects New Delhi's weather. Also, due to increasing pollution levels and global warming, weather conditions are affected by various other factors. This research did not explore the impact of the environmental conditions of surrounding places on the temperature of New Delhi, which is a limitation of this work. This work has not explored and compared results with RNN-based complex models such as Gated Recurrent Unit in RNNs, multivariate long short-term memory, and XGBoost models for temperature predictions, which can also be considered as limitations of this research work. Therefore, a more advanced or hybrid neural network and RNN approach may be explored for prediction a discussion of which is out of the scope of this paper. This is the topic of our future research work.

ACKNOWLEDGMENTS

We thank to all the reviewers for reviewing the manuscript and giving the suggestive comments for improvement of the article.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Virendra Kumar Shrivastava https://orcid.org/0000-0003-4049-2102
Sachi Nandan Mohanty https://orcid.org/0000-0002-4939-0797

REFERENCES

Abrahamsen, E. B., Brastein, O. M., & Lie B. (2018). Machine learning in Python for weather forecast based on freely available weather data, Proceedings of the 59th Conference on Simulation and Modelling, Oslo, Norway. https://doi.org/10.3384/ecp18153169

Anwar, S., Mustafa, M. Nasr, M. A., & Abdulaziz, A. (2017). Predicting surface roughness and exit chipping size in BK7 glass during rotary ultrasonic machining by adaptive neuro-fuzzy inference system (ANFIS), Proceedings of the International Conference on Industrial Engineering and Operations Management, Rabat, Morocco.

Bhatkande, S. S., & Hubballi, R. G. (2016). Weather prediction based on decision tree algorithm using data mining techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, *5*(5), 483–487.

Biau, G. (2012). Analysis of a random forests model. *Journal of Machine Learning Research*, *13*, 1063–1095.

Biswas, S. K., Sinha, N., Purkayastha, B., & Marbaniang, L. (2014). Weather prediction by recurrent neural network dynamics. *International Journal of Intelligent Engineering Informatics*, *2*(2/3), 166–180. https://doi.org/10.1504/IJIEI.2014.066208

Bochenek, B., & Ustrnul, Z. (2022). Machine learning in weather prediction and climate analyses—Applications and perspectives. *Atmosphere*, *13*(2), 180. https://doi.org/10.3390/atmos13020180

Denny, A. P., Francisco, R., Joaquín, T. S., Sergio, T., & Joaquín, H. (2019). Using weather conditions and advanced machine learning methods to predict soccer outcome, AGILE–2019, Limassol.

Dolara, A., Gandelli, A., Grimaccia, F., & Leva, S. (2017). Weather-based machine learning technique for Day-Ahead wind power forecasting, Proceedings of 6th International Conference on Renewal Energy Research and Applications, San Diego, USA pp. 206-209 Nov 5-9, 2017. https://doi.org/10.1109/ICRERA.2017.8191267

Eva, O. (2012). Modelling using polynomial regression. In *Procedia engineering* (pp. 500–506). Elsevier Ltd. https://doi.org/10.1016/j.proeng.2012.09.545

Gadekallu, T. R., Kidwai, B., Sharma, S., Pareek, R., & Karnam, S. (2019). Application of data mining techniques in weather forecasting. In *Sentiment analysis and knowledge discovery in contemporary business* (pp. 162–174). IGI Global. https://doi.org/10.4018/978-1-5225-4999-4.ch010

Geetha, A., & Nasira, G. M. (2017). Time-series modeling and forecasting: Modeling of rainfall prediction using ARIMA model. *International Journal of Society Systems Science*, *8*(4), 361–372. https://doi.org/10.1504/IJSSS.2016.081411

Grover, A., Kapoor, A., & Horvitz, E. (2015). A deep hybrid model for weather forecasting, Proc. of 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. https://doi.org/10.1145/2783258.2783275

Haleem, S. L. A., Sodagudi, S., Althubiti, S. A., Shukla, S. K., Ahmed, M. A., & Chokkalingam, B. (2022). Improving the predictive response using ensemble empirical mode decomposition based soft sensors with auto encoder deep neural network. *Measurement*, *199*, 111308.

Hemalatha, G., Rao, K. S., & Kumar, D. A. (2021). *Weather prediction using advanced MachineLearning techniques* (Vol. 2089, No. 1, p. 012059). Journal of physics: Conference series. IOP Publishing.

Holmstrom, M., Liu, D., & Vo, C. (2016). *Machine learning applied to weather forecasting*. Stanford University.

Hossain, M., Rekabdar, B., Louis, S. J., & Dascalu, S. (2015). Forecasting the weather of Nevada: A deep learning approach. In 2015 international joint conference on neural networks (IJCNN) (pp. 1–6). IEEE. https://doi.org/10.1109/IJCNN.2015.7280812

Htike, K. K. (2018). Predicting rainfall using neural nets. *International Journal of Computational Science and Engineering*, *17*(4), 353–364. https://doi.org/10.1504/IJCSE.2018.096025

Idicula, S. M., & Mohanty, U. C. (2013). Artificial neural network model in prediction of meteorological parameters during premonsoon thunderstorms. *International Journal of Atmospheric Sciences*, *2013*, 1–14. https://doi.org/10.1155/2013/525383

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, *29*(3), 31–44. https://doi.org/10.1109/2.485891

Joaquin, Q. C., Rasmussen, C. E., & Williams, C. K. (2007). Approximation methods for Gaussian process regression. In *Large-scale kernel machines* (pp. 203–223). MIT Press.

Kaggle.com. (2020). Delhi weather data. [online] kaggle.com/mahirkukreja/delhi-weather-data (Accessed on 20 March 2020).

Kapoor, A., Horvitz, Z., Laube, S., & Horvitz, E. (2014). Airplanes aloft as a sensor network for wind forecasting, Proc. of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, pp. 25–33. https://doi.org/10.1109/IPSN.2014.6846738

Khan, M. S., FransCoenen, C. D., Subhieh, E. S., Mariluz, P., & Asun, R. (2015). An intelligent process model: Predicting spring back in single point incremental forming. *The International*

Journal of Advanced Manufacturing Technology, *76*(9–12), 2071–2082. https://doi.org/10.1007/s00170-014-6431-1

Khan, S. U., Ayub, T., & Rafeeqi, S. F. A. (2013). Prediction of compressive strength of plain concrete confined with ferrocement using artificial neural network (ANN) and comparison with existing mathematical models. *American Journal of Civil Engineering and Architecture*, *1*(1), 7–14. https://doi.org/10.12691/ajcea-1-1-2

Kostas, K., Nikos, K., Cezary, O., & Arkadiusz, S. (2018). *Revisiting urban air quality forecasting: A regression a pproach* (Vol. 5, pp. 177–184). Springer. https://doi.org/10.1007/s40595-018-0113-0

Lee, S., Lee, Y. S., & Son, Y. (2020). Forecasting daily temperatures with different time interval data using deep neural networks. *Applied Sciences*, *10*(5), 1609. https://doi.org/10.3390/app10051609

Liu, Q., Zou, Y., Liu, X., & Linge, N. (2019). A survey on rainfall forecasting using artificial neural network. *International Journal of Embedded Systems*, *11*(2), 240–249. https://doi.org/10.1504/IJES.2019.098300

Madan, S., Kumar, P., Rawat. S., & Choudhury, T. (2018). Analysis of weather prediction using machine learning & big data, IEEE. https://doi.org/10.1109/ICACCE.2018.8441679

Marchuk, G. (2012). *Numerical methods in weather prediction*. Elsevier.

Raed, J., Isam, S., & Ilan, J. (2010). Application of Artificial Neural Networks (ANN) to model the failure of urban water mains. *Mathematical and Computer Modelling*, *51*, 1170–1180. https://doi.org/10.1016/j.mcm.2009.12.033

Sahai, A., Soman, M., & Satyan, V. (2000). All India summer monsoon rainfall prediction using an artificial neural network. *Climate Dynamics*, *16*(4), 291–302. https://doi.org/10.1007/s003820050328

Shad, M., Sharma, Y. D., & Singh, A. (2022). Forecasting of monthly relative humidity in Delhi, India, using SARIMA and ANN models. *Modeling Earth Systems and Environment*, *8*, 1–9. https://doi.org/10.1007/s40808-022-01385-8

Shrivastava, V. K., Kumar, A., Shrivastava, A., Tiwari, A., Thiru, K., & Batra, R. (2021). *Study and trend prediction of Covid-19 cases in India using deep learning techniques* (Vol. 1950, No. 1, p. 012084). Journal of Physics: Conference Series. IOP Publishing.

Shrivastava, V. K., Shrivastava, A., Sharma, N., Mohanty, S. N., & Pattanaik, C. R. (2022). Deep learning model for temperature prediction: an empirical study. *Modeling Earth Systems and Environment*, 1–14. https://doi.org/10.1007/s40808-022-01609-x

Shukla, J., & Mooley, D. (1987). Empirical prediction of the summer monsoon rainfall over India. *Monthly Weather Review*, *115*(3), 695–704. https://doi.org/10.1175/1520-0493(1987)115<0695:EPOTSM>2.0.CO;2

Singhal, A., Phogat, M., Kumar, D., Kumar, A., Dahiya, M., & Shrivastava, V. K. (2022). Study of deep learning techniques for medical image analysis: A review. *Materials Today: Proceedings*, *56*, 209–214. https://doi.org/10.1016/j.matpr.2022.01.071

Voyant, C., Muselli, M., Paoli, C., & Nivet, M. L. (2012). Numerical Weather Prediction (NWP) and hybrid ARMA/ANN model to predict global radiation. *Energy*, *39*(1), 341–355. https://doi.org/10.1016/j.energy.2012.01.006

Wang, D., Lu, Y., Chen, B., & Zhao, Y. (2020). Wind weather prediction based on multi-output least squares support vector regression optimized by bat algorithm. *International Journal of Embedded Systems*, *12*(2), 137–145. https://doi.org/10.1504/IJES.2020.105936

Wang, X., Zaho, Y., & Pourpanah, F. (2020). Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, *11*, 747–750. https://doi.org/10.1007/s13042-020-01096-5

Wongchai, A., Shukla, S. K., Ahmed, M. A., Sakthi, U., & Jagdish, M. (2022). Artificial intelligence-enabled soft sensor and internet of things for sustainable agriculture using ensemble deep learning architecture. *Computers and Electrical Engineering*, *102*, 108128. https://doi.org/10.1016/j.compeleceng.2022.108128

Yalavarthi, R., & Shashi, M. (2009). Atmospheric temperature prediction using support vector machines. *International Journal of Computer Theory and Engineering*, *1*, 55–58.

Yeturu, J. (2019). Analysis of weather data using various regression algorithms. *International Journal of Data Science*, *4*(2), 117–141. https://doi.org/10.1504/IJDS.2019.100321

Zhang, N., Jiang, X., Jing, Z., & Keenan, L. (2018). Gaussian process regression method for classification for high-dimensional data with limited samples, Proc. of 18^{th} International Conference on Information Science and Technology (ICIST), pp. 358–363.

Zhang, X., Mohanty, S. N., Parida, A. K., Pani, S. K., Dong, B., & Cheng, X. (2020). Annual and non-monsoon rainfall prediction modelling using SVR-MLP: An empirical study from Odisha. *IEEE Access*, *8*(1), 30223. https://doi.org/10.1109/ACCESS.2020.2972435

Zhou, K., Zheng, Y., Li, B., Dong, W., & Zhang, X. (2019). Forecasting different types of convective weather: A deep learning approach. *Journal of Meteorological Research*, *33*(5), 797–809. https://doi.org/10.1007/s13351-019-8162-6

AUTHOR BIOGRAPHIES

Dr. Virendra Kumar Shrivastava holds a Doctor of Philosophy in Data Mining and an MTech Honors in Computer Technology Applications. He is the recipient of the European Union Erasmus Mundus research scholarship under the FUSION project and received a Post Doctorate in Research and Training from the University of Lumiere of Lyon2, France, and a CMI Level 5 Certificate in Management and Leadership from the Chartered Management Institute of London. He is a life member of the Computer Society of India and ISTE. Dr. Shrivastava has more than two decades of experience in teaching, research, and academic administration in Indian and UK Universities. Dr. Shrivastava is presently working as a professor at Alliance University, Bangalore. He has published several research papers in reputed international journals and conferences. His research area includes Big Data Analytics, Data Mining, and Machine Learning. He has five patents to his credit.

Aastik Shrivastava has done BTech from International Institute of Information Technology, Bhubaneshwar, Odisha, India. Presently, he is working as a software development engineer at Siemens Healthineers, Bangalore. He is a national finalist of Philips Code 2 Care 2021. He is a machine learning and deep learning enthusiast with a keen interest in the research and development of various prediction and forecasting algorithms for weather, disease, among others. His research includes healthcare data analytics, Big data, and machine learning.

Dr. Nonita Sharma is working as an associate professor, at Indira Gandhi Delhi Technical University for Women, New Delhi. She has more than 15 years of teaching experience. Her major area of interest includes data mining, bioinformatics, time series forecasting, and wireless sensor networks. She has published several papers in International/National Journals/Conferences and book chapters. She received the best paper award for her research paper in Mid-Term Symposium organized by CSIR, Chandigarh. She has been awarded the Best Teacher Award in view of recognition of her contributions, achievements, and excellence in Computer Science and Engineering at NIT Jalandhar. She has been awarded Best Content Guru Award by Infosys twice. She has authored a book titled, "Analysis of Algorithms." She has been the editor of various books published by eminent publishers like WILEY, Taylor & Francis, and CRC Press. She is a member of IEEE and has been shortlisted in the Top 5 for IEEE Women Achiever Award. She is the reviewer of many peer-reviewed journals and contributed to academic research in terms of projects, papers, and patents.

Dr. Sachi Nandan Mohanty received his PostDoc from IIT Kanpur in the year 2019 and PhD from IIT Kharagpur, India, in the year 2015, with an MHRD scholarship from Government of India. He has authored/edited 28 books, published by IEEE-Wiley, Springer, Wiley, CRC Press, NOVA, and DeGruyter. His research areas include Data mining, Big Data Analysis, Cognitive Science, Fuzzy Decision Making, Brain-Computer Interface, Cognition, and Computational Intelligence. Prof. SN Mohanty received four Best Paper Awards during his PhD at IIT Kharagpur from International Conference in Beijing, China, and the other at International Conference on Soft Computing Applications organized by IIT Roorkee in the year 2013. He has awarded the Best Thesis Award, the first prize by the Computer Society of India in the year 2015. He has guided nine PhD scholars. He has published 120 International Journals of International repute and has been elected as Fellow of the Institute of Engineers, European Alliance Innovation (EAI) Springer, and a Senior member of the IEEE Computer Society Hyderabad chapter. He is also the reviewer of the Journal of Robotics and Autonomous Systems (Elsevier), Computational and Structural Biotechnology Journal (Elsevier), Artificial Intelligence Review (Springer), and Spatial Information Research (Springer).

Dr. Chinmaya Ranjan Pattanaik has received his PhD in Computer Science and Engineering from KIIT University in 2016. He has published in 16 international journals and has six patents. Presently, he is working as an associate professor in the department of Computer Science and Engineering at Ajay Binay Institute of Technology Cuttack under BPUT Rourkela. He is also a life member of ISTE.

How to cite this article: Shrivastava, V. K., Shrivastava, A., Sharma, N., Mohanty, S. N., & Pattanaik, C. R. (2023). Deep learning model for temperature prediction: A case study in New Delhi. *Journal of Forecasting*, 1–16. https://doi.org/10.1002/for.2966