

A framework for cloud cover prediction using machine learning with data imputation

Nabanita Mandal, Tanuja Sarode

Computer Engineering Department, Thadomal Shahani Engineering College, Affiliated to the University of Mumbai, Mumbai, India

Article Info

Article history:

Received Jun 22, 2023

Revised Jul 12, 2023

Accepted Jul 17, 2023

Keywords:

Climate prediction

Cloud cover

Data imputation

Deep learning

Machine learning

ABSTRACT

The climatic conditions of a region are affected by multiple factors. These factors are dew point temperature, humidity, wind speed, and wind direction. These factors are closely related to each other. In this paper, the correlation between these factors is studied and an approach has been proposed for data imputation. The idea is to utilize all these features to obtain the prediction of the total cloud cover of a region instead of removing the missing values. Total cloud cover prediction is significant because it affects the agriculture, aviation, and energy sectors. Based on the imputed data which is obtained as the output of the proposed method, a machine learning-based model is proposed. The foundation of this proposed model is the bi-directional approach of the long short-term memory (LSTM) model. It is trained for 8 stations for two different approaches. In the first approach, 80% of the entire data is considered for training and 20% of the data is considered for testing. In the second approach, 90% of the entire data is accounted for training and 10% of the data is accounted for testing. It is observed that in the first approach, the model gives less error for prediction.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Nabanita Mandal

Computer Engineering Department, Thadomal Shahani Engineering College, Affiliated to the University of Mumbai

Bandra (w), Mumbai, Maharashtra, India

Email: nabanita.mandal@thadomal.org

1. INTRODUCTION

The climate of a region directly affects human life in many ways. The significance of predicting climate parameters like wind speed, dew point temperature, and cloud cover helps us in determining and planning crop yield. The energy and aviation sectors also benefitted from the predictions. Cloud cover of a place helps predict the rainfall and sunshine duration which leads to better planning of solar energy initiatives. The cloud cover also affects visibility which is a very significant factor for airline operations. The prediction of cloud cover is influenced by various factors like rainfall, wind speed, and direction, and vapor pressure. The unit of measurement of cloud cover is oktas.

In Maharashtra, the climate varies significantly from one region to another. The different regions of Maharashtra are Vidarbha, Marathwada, Konkan, and Madhya Maharashtra. The state experiences heavy cloud cover in some parts during certain months and also experiences low to negligible cloud cover during other months in different regions. Since agriculture is the main occupation of Maharashtra, it is important to predict the cloud cover so that rainfall can be determined and proper water management can be done. Some regions experience severe drought conditions due to lack of rainfall whereas some regions experience floods due to excessive rainfall [1]. Over the last few years, machine learning techniques have become widely popular due to their application to a large number of environmental causes [2]. It is used for the prediction of

drought, and rainfall prediction [3], [4]. Water level prediction can also be done with good accuracy using deep learning algorithms [5]. The amount of rainfall a particular region receives can be decided by predicting the cloud cover of that region.

Cloud cover determines how much the sun is obstructed by clouds. This helps in solar plant operation. Cloud motion vectors are used for forecasting solar radiation [6]. Tracking of the cloud is done using binary cross-correlation. Along with this, the maximum cross-correlation technique is utilized. Quality control is done by measuring the vectors for incorrectly detected motions. Assessment of cloud cover is also done in numerical weather prediction by researchers [7]. A new facility was introduced which conceals the sunburn effect present in the background. Detection of thin clouds is better when using this method with artificial neural network (ANN) [8]. To determine the cloud over a particular region, satellite images are also used. The data set created uses satellite images for the classification of cloud patches [9]. Convolution neural network (CNN) along with data augmentation and regularization was used. The satellite images were also used for predicting the movement of clouds using neural networks [10]. Further deep-learning techniques are also used to classify cloudy or clear skies using images [11]. Various deep-learning techniques are applied to these satellite images for forecasting [12]–[15].

This paper consists of five sections. Section 2 consists of the proposed framework. It describes the proposed data imputation method and the proposed model. Section 3 is the method section which includes the dataset details and data handling techniques. Section 4 describes the results and discussion which consists of the tables and their description. Section 5 is the conclusion section which describes the summary of the overall work done in this research.

2. PROPOSED FRAMEWORK

The proposed framework is based on machine learning which emphasizes learning by identifying the pattern of the data. The framework consists of data imputation and model building. Figure 1 shows the proposed framework. The proposed data imputation method eliminates the need for deletion of rows containing missing values thereby preserving the size of the dataset. It uses iterative and k-nearest neighbors (KNN) imputation. The proposed model is deep learning based which uses the principle of long short-term memory. It consists of small memory units that can handle data in the time series format and also pass the information bi-directionally. The idea behind this proposed framework is to utilize the entire data that is available and build a model that gives less prediction error.

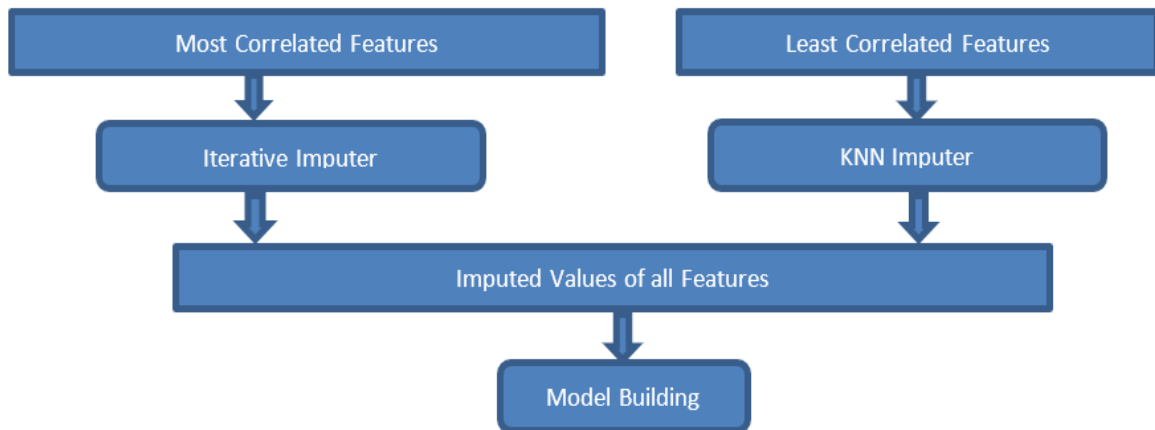


Figure 1. Proposed framework

3. METHOD

To obtain the prediction, in this research the dataset has been acquired from the India Meteorological Department, Pune, Maharashtra, India [16]. The dataset consists of 8 stations namely: Akola, Nagpur, Chikalthana, Parbhani, Colaba, Ratnagiri, Kolhapur, and Parbhani. The different features that are included here are mean sea level pressure (MMSLP), mean dew point temperature (MDPT), mean relative humidity (MRH), mean vapor pressure (MVP), mean total cloud (MTC), total mean rainfall (TMRF), mean wind speed (MWSP) and the directions of wind: north (N), south (S), north east (NE), south east (SE), east (E), west (W), south west (SW), north west (NW). The raw data is analyzed and it is

found that there exist some missing values. These are handled using data imputation techniques which are described in the next section. The resultant is the imputed data which is further scaled. The necessity of scaling arises because different features are measured in different units. Some of them are measured in percentage, millimeters, and kilometers per hour. The data then further needs to be trained and tested. The machine learning model takes this time series data as input [17], [18]. Figure 2 represents the steps followed for handling the data.



Figure 2. Data handling

3.1. Data imputation

The missing values are predicted using data imputation techniques [19]–[25]. To predict the mean total cloud cover of a region, it is important to understand how the features influence the target variable. In this proposed data imputation method, this influence is studied by understanding the correlation between them [26]. To determine the correlation, the concept of heatmap is used [27]. A heatmap is a visual representation of the different features. It uses a color-coding scheme to show the correlation. The most correlated features are combined into one group and the less correlated features are put in another group as mentioned in Figure 1. In the proposed method, iterative imputer is applied to features of the first group whereas the second group of features uses KNN imputer [28]. Table 1 represents these features for different stations. It is observed that for all 8 stations, there are some common most correlated and some common least correlated features.

Table 1. Most and least correlated features

Station	Most correlated features	Least correlated features
Ratnagiri	MDPT, MRH, MVP, MTC, MWSP, MMSLP, TMRF, W, SW, N, E, SE, NW	S, NE
Akola	TMRF, MVP, MRH, MTC, MDPT, MMSLP, MWSP, W, SW, E, NE, NW	N, S, SE
Chikalthana	MMSLP, MDPT, MRH, MVP, MTC, MWSP, TMRF, W, SW, N, E, NE, SE, NW	S
Parbhani	MDPT, MRH, MVP, TMRF, MWSP, MTC, MMSLP, W, SW, E, NE, NW	N, S, SE
Kolhapur	MDPT, MRH, MVP, MTC, TMRF, MMSLP, MWSP, W, SW, E, NE, SE	N, S, NW
Nashik	MDPT, MRH, MVP, TMRF, MMSLP, MTC, MWSP, W, SW, E, NW, SE, NE	N, S
Colaba	MDPT, MRH, MVP, TMRF, MMSLP, MTC, MWSP, W, SW, N, E, NE, SE, NW	S
Nagpur	MDPT, MRH, MVP, TMRF, MMSLP, MTC, N, E, W, SW, NW, NE, SE	S

3.2. Model building

In this proposed model, first, the data needs to be arranged in a time series format. The conversion of data into time series is an important aspect here. Time step, feature, and batch size are used as input, and output is based on values of features at previous timesteps along with current state values. Figure 3 represents the proposed model. The basis of the proposed model is bi-directional long short-term memory (LSTM) of deep learning [29]–[32]. The benefit of using this model is that the information stored in the cell is used for future processing. It is a bidirectional model where the processing is sequential and consists of two LSTMs: one will take the input in the forward direction and the other will take it in the backward direction. Training of both LSTM models is done considering the training testing split for 80%-20% and 90%-10%.

The number of layers used in this proposed model are 3 and to validate the results, mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) are used [33], [34]. The proposed model is a sequential model created by stacking each layer one by one. There are 3 layers: two bi-directional layers and a dense layer. The first bi-directional layer allows the simultaneous processing of input sequences in both directions. The output of this layer goes to the second bi-directional layer which again processes it in both directions. The dense layer is the last layer which gives a single output for each input.

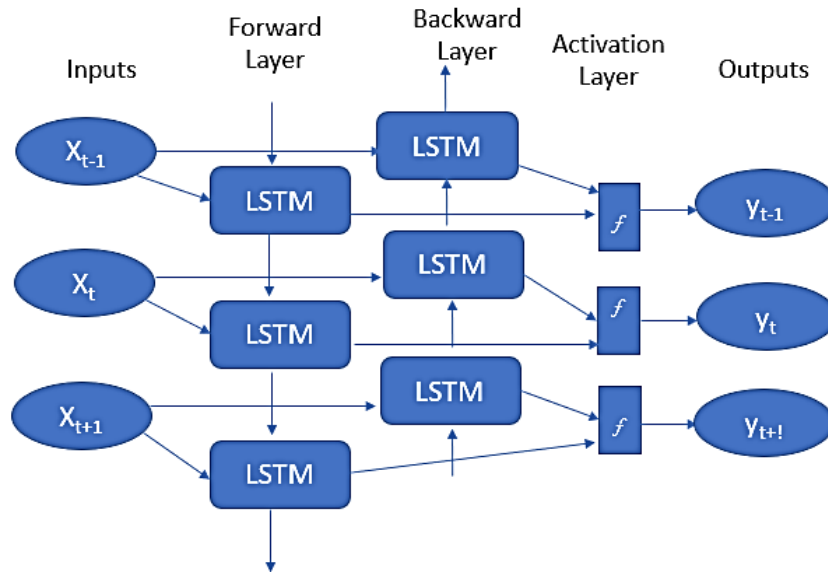


Figure 3. Proposed model

4. RESULTS AND DISCUSSION

The model is trained for various epochs: 100, 200, 300, 500, 800, and 1,000 for two different approaches. In the first approach, the model is trained by taking 80% of the data and tested using the remaining 20% of the data. In the second approach, the model is trained by taking 90% of the data and tested on 10% of the data. The model performance is evaluated by considering the MSE, RMSE, and MAE.

4.1. Model performance evaluation using MSE

For the evaluation of the models, the mean squared error is used. In (1) represents the average of values of the squared difference between actual (y_i) and predicted values (\hat{y}_i).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{1}$$

Table 2 represents the MSE values of different stations for different epochs considering the first approach where the model training for 80% of data and testing of 20% of data. It is observed that the model gives less MSE value for 500 epochs for Akola, Chikalthana, and Kolhapur. For Ratnagiri and Nagpur, 100 epochs give less MSE. For Parbhani, Nashik, and Colaba the MSE values are less in the 200, 800, and 1,000 epochs respectively.

Table 2. MSE values for 80% training data-20% testing data

Epochs	100	200	300	500	800	1000
Ratnagiri	0.0276733	0.0276801	0.0285312	0.0289139	0.0293265	0.0290827
Akola	0.0005606	0.0005514	0.0005319	0.0005153	0.0005424	0.0005555
Chikalthana	0.0290094	0.0302561	0.0288897	0.0288534	0.0298026	0.0302239
Parbhani	0.0010222	0.0009553	0.0010082	0.0009991	0.0010242	0.0009699
Kolhapur	0.0017716	0.0017656	0.0017424	0.0017085	0.0017338	0.0017163
Nashik	0.0070312	0.0069630	0.0068431	0.0066161	0.0066138	0.0065996
Colaba	0.0013896	0.0011890	0.0012261	0.0012615	0.0012140	0.0011141
Nagpur	0.0120946	0.0136935	0.0129138	0.0146175	0.0132889	0.0146835

Table 3 represents the MSE values for the second approach when the model is trained considering 90% data and tested at 10% data. The MSE values are less than 100 epochs for Ratnagiri, Akola, and Nagpur stations. For Chikalthana station, 500 and for Colaba station 300 epochs give less MSE. For 200 epochs Parbhani, Kolhapur and Nashik stations give less MSE values. Table 4 shows which approach gives the least MSE values and the epoch number. Except for Parbhani and Colaba stations, all stations give good predictions for the first approach when 80% of data is used for training.

Table 3. MSE values for 90% training data-10% testing data

Epochs	100	200	300	500	800	1,000
Ratnagiri	0.0291420	0.0294234	0.0305199	0.0302947	0.0310176	0.0306762
Akola	0.0005312	0.0006283	0.0005633	0.0005318	0.0006953	0.0007747
Chikalthana	0.0333911	0.0348839	0.0329928	0.0327869	0.0332216	0.0341873
Parbhani	0.0009351	0.0008068	0.0009174	0.0008620	0.0008469	0.0008114
Kolhapur	0.0021738	0.0020500	0.0020541	0.0021518	0.0021635	0.0021426
Nashik	0.0129962	0.0118727	0.0125976	0.0121322	0.0119440	0.0119992
Colaba	0.0012000	0.0010356	0.0010237	0.0010360	0.0010376	0.0011377
Nagpur	0.0139544	0.0155561	0.0146445	0.0144931	0.0151378	0.0162803

Table 4. Least MSE values

Stations	Least MSE value	Training-Testing Split	Epoch Number
Ratnagiri	0.0276733	80%-20%	100
Akola	0.0005153	80%-20%	500
Chikalthana	0.0288534	80%-20%	500
Parbhani	0.0008068	90%-10%	200
Kolhapur	0.0017085	80%-20%	500
Nashik	0.0066138	80%-20%	800
Colaba	0.0010237	90%-10%	300
Nagpur	0.0120946	80%-20%	100

4.2. Model performance evaluation using RMSE

The root mean squared error is represented by (2). The RMSE values by changing the number of epochs is shown in Table 5. The model is trained considering 80% of data and tested at the remaining 20% of data.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

Ratnagiri and Nagpur stations show less RMSE value for 100 epochs. For 500 epochs, Akola, Chikalthana and Kolhapur stations show less RMSE. Parbhani station shows less RMSE for 200 epochs, Nashik station shows less RMSE value for 800 epochs and Colaba station shows less RMSE for 1,000 epochs. Table 6 depicts the RMSE values for 90% of training data and 10% of testing data. RMSE values are less for 100 epochs for Ratnagiri, Akola, and Nagpur. For Chikalthana station 500 epochs and for Colaba station 300 epochs give less RMSE. For 200 epochs, low RMSE values are shown for Parbhani, Kolhapur, and Nashik stations. Table 7 shows the epoch numbers and the approach where the least RMSE values are obtained. Colaba and Parbhani stations give low RMSE values for the second approach where 90% of data is used for training, the rest of the stations give less RMSE for the first approach where 80% of data is used as training data.

Table 5. RMSE values for 80% training data-20% testing data

Epochs	100	200	300	500	800	1000
Ratnagiri	0.1663531	0.1663734	0.1689120	0.1700409	0.1712499	0.1705367
Akola	0.0236785	0.0234837	0.0230634	0.0227012	0.0232896	0.0235708
Chikalthana	0.1703214	0.1739428	0.1699698	0.1698631	0.1726345	0.1738504
Parbhani	0.0319722	0.0309094	0.0317524	0.0316090	0.0320033	0.0311436
Kolhapur	0.0420904	0.0420197	0.0417429	0.0413348	0.0416389	0.0414293
Nashik	0.0838524	0.0834446	0.0827233	0.0813397	0.0812258	0.0812381
Colaba	0.0372777	0.0344825	0.0350163	0.0355188	0.0348427	0.0333789
Nagpur	0.1099754	0.1170193	0.1136392	0.1209029	0.1152775	0.1211758

Table 6. RMSE values for 90% training data-10% testing data

Epochs	100	200	300	500	800	1000
Ratnagiri	0.1707103	0.1715326	0.1746996	0.1740538	0.1761183	0.1751462
Akola	0.0230494	0.0250678	0.0237359	0.0230612	0.0263686	0.0278347
Chikalthana	0.1827323	0.1867723	0.1816392	0.1810715	0.1822681	0.1848983
Parbhani	0.0305805	0.0284058	0.0302889	0.0293614	0.0291015	0.0284862
Kolhapur	0.0466248	0.0452776	0.0453232	0.0463883	0.0465137	0.0462883
Nashik	0.1140010	0.1089621	0.1122392	0.1101463	0.1092887	0.1095409
Colaba	0.0346411	0.0321809	0.0319963	0.0321879	0.0322122	0.0337305
Nagpur	0.1181291	0.1247243	0.1210148	0.1203875	0.1230361	0.1275946

Table 7. Least RMSE values

Station	Least RMSE value	Training-testing split	Epoch number
Ratnagiri	0.1663531	80%-20%	100
Akola	0.0227012	80%-20%	500
Chikalthana	0.1698631	80%-20%	500
Parbhani	0.0284058	90%-10%	200
Kolhapur	0.0413348	80%-20%	500
Nashik	0.0812258	80%-20%	800
Colaba	0.0319963	90%-10%	300
Nagpur	0.1099754	80%-20%	100

4.3. Model performance evaluation using MAE

The mean absolute error is described using (3). The MAE values are shown for different stations in Table 8, 80% of the entire training set is taken for training and the remaining 20% is taken for data testing.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}|^2 \quad (3)$$

Less MAE values are obtained for 500 epochs for Kolhapur and Nashik stations. For 100 epochs Chikalthana, Parbhani, and Nagpur give less MAE values. For 300 epochs, Akola station gives less MAE. Colaba and Ratnagiri station gives less MAE for 1,000 epochs. Table 9 shows the MAE values when 90% of training data is considered and 10% of testing data is considered. MAE values are less for 200 epochs for Parbhani and Kolhapur stations. For 100 epochs less MAE value is obtained for the Akola region. For 500 epochs Nashik, Colaba, and Nagpur regions have less MAE. For Ratnagiri and Chikalthana stations, less MAE is obtained for 800 epochs. Table 10 shows the epoch numbers where the least MAE values are obtained and also the approach used. Except for the Akola station, all stations give less MAE values when 80% of data is used for training.

Table 8. MAE values for 80% training data-20% testing data

Epochs	100	200	300	500	800	1,000
Ratnagiri	0.0689868	0.0678983	0.0660602	0.0655668	0.0657325	0.0644902
Akola	0.0142865	0.0148944	0.0135877	0.0175612	0.0140383	0.0151419
Chikalthana	0.1288641	0.1295586	0.1333525	0.1310643	0.1308458	0.1290380
Parbhani	0.0220812	0.0237574	0.0238507	0.0239057	0.0249565	0.0241586
Kolhapur	0.0303300	0.0303221	0.0299984	0.0290924	0.0303035	0.0298363
Nashik	0.0379222	0.0383146	0.0378659	0.0376852	0.0384562	0.0381444
Colaba	0.0267135	0.0254817	0.0246748	0.0255479	0.0255461	0.0244953
Nagpur	0.0689747	0.0822926	0.0806480	0.0913560	0.0731328	0.0899981

Table 9. MAE values for 90% training data-10% testing data

Epochs	100	200	300	500	800	1000
Ratnagiri	0.0755583	0.0731435	0.0696919	0.0707707	0.0673403	0.0704220
Akola	0.0129672	0.0184834	0.0154448	0.013813	0.0210741	0.0233999
Chikalthana	0.1413201	0.1419384	0.1448822	0.1406412	0.140016	0.1434573
Parbhani	0.0231625	0.0221151	0.0232333	0.0227556	0.0224511	0.0222715
Kolhapur	0.0356100	0.0332673	0.0336692	0.0360381	0.0358249	0.0352319
Nashik	0.0504187	0.0484206	0.0494217	0.0478641	0.0489036	0.0489545
Colaba	0.0263760	0.0251136	0.0222471	0.0247312	0.0246646	0.0267234
Nagpur	0.0680398	0.08949838	0.0842604	0.0707163	0.0841946	0.0962322

Table 10. Least MAE values

Station	Least MAE value	Training-Testing Split	Epoch Number
Ratnagiri	0.0644902	80%-20%	1,000
Akola	0.0129672	90%-10%	100
Chikalthana	0.1288641	80%-20%	100
Parbhani	0.0220812	80%-20%	100
Kolhapur	0.0290924	80%-20%	500
Nashik	0.0376852	80%-20%	500
Colaba	0.0244953	80%-20%	1000
Nagpur	0.0689747	80%-20%	100

5. CONCLUSION

The machine learning-based model used to predict the cloud cover in different regions of Maharashtra depends immensely on the data because it is used for learning the pattern. The correlation-based data imputation method proposed here gives the most correlated and least correlated features. The direction of the wind in south (S) is the least correlated feature which is common to all the 8 stations. The remaining features apart from certain wind directions are mostly correlated to each other for all the stations. Iterative imputer replaces the missing values by repeatedly iterating over the most correlated features and KNN imputer uses least correlated features to replace the missing values using it is nearest neighbors. The imputed data is scaled and split for training and testing. The model is built by considering a two-way approach to information flow. The model runs for multiple epochs to give the least MSE, RMSE, and MAE values. The comparison is done for 8 stations based on the two approaches used. It is observed that the model trained at 80% and tested at 20% has the least MAE, RMSE, and MSE values for most of the stations as compared to the model trained at 90% and 10%.

ACKNOWLEDGEMENTS

The data is acquired from the India Meteorological Department, Pune, Maharashtra, India. The surface data supplied by them is utilized here. We are thankful to them for the dataset. It helped us conduct this research efficiently.




REFERENCES

- [1] R. N. Singh, A. Chaudhary, H. Pathak, J. Rane, and S. Potekar, "Climatic trends in western Maharashtra, India," National Institute of Abiotic Stress Management, 2020.
- [2] A.-L. Balogun and N. Adebisi, "Sea level prediction using ARIMA, SVR and LSTM neural network: assessing the impact of ensemble Ocean-Atmospheric processes on models' accuracy," *Geomatics, Natural Hazards and Risk*, vol. 12, no. 1, pp. 653–674, Jan. 2021, doi: 10.1080/19475705.2021.1887372.
- [3] W. Almikaeel, L. Čubánová, and A. Šoltész, "Hydrological drought forecasting using machine learning—gidra river case study," *Water*, vol. 14, no. 3, Jan. 2022, doi: 10.3390/w14030387.
- [4] S. Hudnurkar and N. Rayavarapu, "On the performance analysis of rainfall prediction using mutual information with artificial neural network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 2, pp. 2101–2113, Apr. 2023, doi: 10.11591/ijece.v13i2.pp2101-2113.
- [5] Z. Xie, Q. Liu, and Y. Cao, "Hybrid deep learning modeling for water level prediction in yangtze river," *Intelligent Automation and Soft Computing*, vol. 28, no. 1, pp. 153–166, 2021, doi: 10.32604/iasc.2021.016246.
- [6] F. J. Batlles, J. Alonso, and G. López, "Cloud cover forecasting from METEOSAT data," *Energy Procedia*, vol. 57, pp. 1317–1326, 2014, doi: 10.1016/j.egypro.2014.10.122.
- [7] K. D. Hutchison, B. D. Isager, and X. Jiang, "Quantitatively assessing cloud cover fraction in numerical weather prediction and climate models," *Remote Sensing Letters*, vol. 8, no. 8, pp. 723–732, Aug. 2017, doi: 10.1080/2150704X.2017.1317932.
- [8] M. Krinitskiy, "Cloud cover estimation optical package: New facility, algorithms and techniques," *AIP Conference Proceedings*, 2017, doi: 10.1063/1.4975540.
- [9] E. J. Rhee, "A deep learning approach for classification of cloud image patches on small datasets," *Journal of information and communication convergence engineering*, vol. 16, no. 3, pp. 173–178, 2018.
- [10] M. Penteliuc and M. Frincu, "Prediction of cloud movement from satellite images using neural networks," *2019 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 222–229, Sep. 2019, doi: 10.1109/SYNASC49474.2019.00038.
- [11] M. Kalkan *et al.*, "Cloudy/clear weather classification using deep learning techniques with cloud images," *Computers and Electrical Engineering*, vol. 102, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108271.
- [12] V.-S. Ionescu, G. Czubala, and E. Mihuleț, "DeePS at: A deep learning model for prediction of satellite images for nowcasting purposes," *Procedia Computer Science*, vol. 192, pp. 622–631, 2021, doi: 10.1016/j.procs.2021.08.064.
- [13] P. Catalán-Valdelomar, J. L. Gómez-Amo, F. Scarlatti, C. Peris-Ferrús, and M. P. Utrillas, "Comparison of two different techniques to determine the cloud cover from all-sky imagery," in *Remote Sensing of Clouds and the Atmosphere XXVI*, Sep. 2021, p. 30, doi: 10.1117/12.2599517.
- [14] Y. Son, Y. Yoon, J. Cho, and S. Choi, "Cloud cover forecast based on correlation analysis on satellite images for short-term photovoltaic power forecasting," *Sustainability*, vol. 14, no. 8, Apr. 2022, doi: 10.3390/su14084427.
- [15] I. Bandara, L. Zhang, and K. Mistry, "Deep learning based short-term total cloud cover forecasting," in *2022 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892773.
- [16] "Climate Research & Services, Pune," India Meteorological Department. <https://www.imdpune.gov.in/> (accessed Jan. 15, 2023).
- [17] M. Bilgili, A. İlhan, and Ş. Ünal, "Time-series prediction of hourly atmospheric pressure using ANFIS and LSTM approaches," *Neural Computing and Applications*, vol. 34, no. 18, pp. 15633–15648, Sep. 2022, doi: 10.1007/s00521-022-07275-5.
- [18] E. Antony, N. S. Sreekanth, R. K. S. Kumar, and N. T., "Data preprocessing techniques for handling time series data for environmental science studies," *International Journal of Engineering Trends and Technology*, vol. 69, no. 5, pp. 196–207, May 2021, doi: 10.14445/22315381/IJETT-V69I5P227.
- [19] N. Umar and A. Gray, "Comparing single and multiple imputation approaches for missing values in univariate and multivariate water level data," *Water*, vol. 15, no. 8, Apr. 2023, doi: 10.3390/w15081519.
- [20] S. Zhang, "Nearest neighbor selection for iteratively kNN imputation," *Journal of Systems and Software*, vol. 85, no. 11, pp. 2541–2552, Nov. 2012, doi: 10.1016/j.jss.2012.05.073.
- [21] N. Fazakis, G. Kostopoulos, S. Kotsiantis, and I. Mporas, "Iterative robust semi-supervised missing data imputation," *IEEE*




- Access, vol. 8, pp. 90555–90569, 2020, doi: 10.1109/ACCESS.2020.2994033.
- [22] M. M. Mijwil, A. W. Abdulqader, S. M. Ali, and A. T. Sadiq, “Null-values imputation using different modification random forest algorithm,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 1, pp. 374–383, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp374-383.
- [23] G. S. Hassan, N. J. Ali, A. K. Abdulsahib, F. J. Mohammed, and H. M. Ghenni, “A missing data imputation method based on salp swarm algorithm for diabetes disease,” *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 3, pp. 1700–1710, Jun. 2023, doi: 10.11591/eei.v12i3.4528.
- [24] J. M. Z. Hoque, J. Hossen, S. Sayeed, C. M. Tawsif K., J. Ganesan, and J. E. Raja, “Automatic missing value imputation for cleaning phase of diabetic’s readmission prediction model,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 2001–2013, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2001-2013.
- [25] G. Madhu and G. Nagachandrika, “A new paradigm for development of data imputation approach for missing value estimation,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 6, pp. 3222–3228, Dec. 2016, doi: 10.11591/ijece.v6i6.pp3222-3228.
- [26] S. Chormunge and S. Jena, “Correlation based feature selection with clustering for high dimensional data,” *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 542–549, Dec. 2018, doi: 10.1016/j.jesit.2017.06.004.
- [27] L. Wilkinson and M. Friendly, “The history of the cluster heat map,” *The American Statistician*, vol. 63, no. 2, pp. 179–184, May 2009, doi: 10.1198/tas.2009.0033.
- [28] L. Muflikhah, N. Hidayat, and D. J. Hariyanto, “Prediction of hypertension drug therapy response using K-NN imputation and SVM algorithm,” *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 15, no. 1, pp. 460–467, Jul. 2019, doi: 10.11591/ijeecs.v15.i1.pp460-467.
- [29] S. Liang, D. Wang, J. Wu, R. Wang, and R. Wang, “Method of bidirectional LSTM modelling for the atmospheric temperature,” *Intelligent Automation and Soft Computing*, vol. 29, no. 3, pp. 701–714, 2021, doi: 10.32604/iasc.2021.020010.
- [30] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “The performance of LSTM and BiLSTM in forecasting time series,” in *2019 IEEE International Conference on Big Data (Big Data)*, Dec. 2019, pp. 3285–3292, doi: 10.1109/BigData47090.2019.9005997.
- [31] Q. Li, Y. Zhao, and F. Yu, “A novel multichannel long short-term memory method with time series for soil temperature modeling,” *IEEE Access*, vol. 8, pp. 182026–182043, 2020, doi: 10.1109/ACCESS.2020.3028995.
- [32] T. Bhandarkar, V. K. N. Satish, S. Sridhar, R. Sivakumar, and S. Ghosh, “Earthquake trend prediction using long short-term memory RNN,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1304–1312, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1304-1312.
- [33] T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not,” *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.
- [34] C. Jittawiriyankoon, “Estimation of regression-based model with bulk noisy data,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 3649–3656, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3649-3656.

BIOGRAPHIES OF AUTHORS



Nabanita Mandal    is pursuing Ph.D. from Thadomal Shahani Engineering College, affiliated with the University of Mumbai, India. She has also done her master’s in engineering from the same institution in 2014 in computer engineering. She is currently working as an Assistant Professor in Computer Engineering Department at Thadomal Shahani Engineering College, Mumbai, India since 2013. Her research area includes climate data analysis, machine learning, deep learning, graph theory, and system security. She has published 4 papers in international journals and conferences. She can be contacted at email: nabanita.mandal@thadomal.org.



Tanuja Sarode    received Ph.D. in engineering from Mukesh Patel School of Technology Management and Engineering (MPSTME), NMIMS University in 2010. She is a Professor and Head of the Computer Engineering Department at Thadomal Shahani Engineering College, Mumbai, India. She has received the best paper award for SPICON-2022 International Conference. Her research interests are image processing, artificial intelligence, and machine learning. She has more than 300 publications in international journals and conferences. Under her guidance, many students have already completed their Ph.D. and many are enrolled. She is also a Ph.D. reviewer at MPSTME, NMIMS University, Mumbai, India. She can be contacted at email: tanuja.sarode@thadomal.org.