Optimized Hybrid Neural Network for Wind Speed Forecasting

T. M. Rubaith Bashar*, Mohammad Munem†, MD. Safayet Islam‡, Murad Hossain§, Tasnim Binte Shawkat¶ and Habibur Rahaman‖
Department of Electrical & Computer Engineering
Rajshahi University of Engineering & Technology, Bangladesh*†‡§¶
Memorial University of Newfoundland, Canada‖
*rubaithbashar@gmail.com**, *munem.mh@gmail.com*†, *ornob39@gmail.com*‡, *murad2771@gmail.com*§,
m.shawkat.bd@ieee.org¶ and *mhrahaman@mun.ca*‖

***Abstract*—Though wind power capacity all over the world is increasing rapidly, the availability of wind power generation mostly reclines on wind speed, which is a random variable with stochastic nature. Therefore, robust technique with powerful feature extraction capability is required to predict wind speed accurately. In this paper, we have recommended a hybrid model using convolutional neural network (CNN) and long-short term memory (LSTM). where CNN is used for extracting fuzzy input features and LSTM to catch the sequence to predict wind speed accurately. As deep learning models are associated with multiple hyper-parameters with great impact, Bayesian optimization algorithm is used for hyper-parameter tuning. Additionally, the performance of some established machine learning models are added on the same data-set. It is observed that, the proposed Bayesian optimized CNN-LSTM hybrid model surpasses the other four established models like SVM, ANN, CNN and LSTM in terms of different performance evaluation metrics like mean absolute error, root mean error and root mean square error.**

***Index Terms*—wind speed forecasting, Deep learning, Hybrid neural network**

I. Introduction

Fossil fuels and the other leading resources of energy are non-renewable. These resources are limited and as a result, the quantity of these resources is dropping rapidly. Added to that, non-renewable energy resources have a minimal impact on the environment and human health [1]. Because of these, scientists and policymakers are emphasizing renewable energy resources. They are working on a variety of instruments to motivate the utilization of several alternative energy resources. Therefore, non-polluting multiple renewable energy resources like wind energy are getting more importance throughout the world. The cost of electricity generated from wind power has reduced to one-seventh of the price in the early 1980s [2]. Hence, adding to the growth in energy demand all over the world, environmental issues and the cost of fossil fuel flourishing in many developing parts of the world are making natural renewable resources like wind energy a competitive option. The advancement in wind energy has been remarkably, forming 1.2 million jobs in the wind energy sector in 2018 [3]. With the installation of about 94 GW of wind power, 2021 was the biggest year yet for the worldwide wind industry. By this year-to-year growth of 53% total worldwide wind power capacity surpass 743 GW [3]. But there is a lack of great research and study on sustainable transition, transformation processes, wind energy conversion systems, wind energy management systems, wind speed, and power production forecasting. As wind speed conducts a vital role in windmill productivity and any kind of unexpected variation in the wind speed affects the whole wind turbine system. So, developing wind speed forecasting has a great financial effect, and it helps the operators to reduce the unreliability of the electrical supply system [4].

In this paper, a hybrid model using a convolution neural network and long-short term memory featuring a Bayesian optimization algorithm is proposed for half-hourly ahead wind speed prediction in Dhaka, Bangladesh using historical meteorological data provided by B.M.D. [5].

II. Related works

Three groups of frequently used techniques can be distinguished as Physical, numerical, and machine learning approaches. Machine learning (ML) techniques rely on historical data that is computationally challenging, yet this technique is the most effective and well-liked because it can identify nonlinear patterns and hence provide results that are more accurate than those from other techniques.

Several basic machine learning models like regression trees, random forest, and support vector machines have shown their satisfactory performance. To forecast the hourly average wind speed in Spain, Troncoso et al. suggested several regression tree models that have a very limited forecasting horizon [6]. In 2017, Lahouar et al. presented hourly wind power forecasting by using a random forest model using 6, 12, and 18 inputs for the wind farm in Tunisia [7]. Zhou et al. proposed a support vector machine method for short-term wind speed prediction [8]. The work was evaluated with different hyperparameter functions to find out the most appropriate functions or values. For predicting more accurately, other models such as the autoregressive moving average (A.R.M.A.) model based on time series data were proposed to predict the hourly wind speed in Spain [9].

Naive ML models with such approaches reach the upper limits as the data resolution continuously increases and the feature dimension is further expanded [10]. To resolve this setback, a good number of deep learning (DL) models have been largely adopted during the last few years. These days, some advancements in neural networks (NN) have demonstrated greater superiority to other model architectures. One such example is the artificial neural network (ANN), which is based on connecting basic processing units known as neurons and is an information technology pattern inspired by the biological nervous system. In 2020, Bayesian optimized ANN models were proposed for short-term wind speed prediction in Newfoundland, Canada. A simple ANN architecture was proposed which is capable to perform precisely because of its hyperparameter optimized nature [11]. The convolutional neural network (CNN) can capture the fuzzy features because of its convolution layer than classical ANN. As a result, the convolutional neural network (CNN) can be applied successfully in wind speed forecasting. In 2021, a 3-CNN model was introduced to forecast wind speed in China [12]. The model can perform wind speed prediction using spatio-temporal feature information. The performance of the model was verified using SCADA data. As wind speed has time sequence characteristics, different recurrent neural networks have been used for this problem. In 2022, Yan et al. proposed a method where the non-linear features were decomposed using EEMD. Then the processed data were fed into the LSTM network [13]. Different hybrid models have been proposed for their diverse characteristics too. In 2022, a hybrid CNN-LSTM model was introduced for wind speed prediction in sailboats [14]. But better performance can be produced by evaluating different hyperparameters associated with the model.

III. Study area and Data collection

The real-time data for this study were collected from Bangladesh meteorological department [5] for Shahjalal international airport, Dhaka (23.8434° N, 90.4029° E). The half-hourly data are taken for the month of June and July of 2013 to 2022. The dataset consists of a total of 29,280 samples including 12 attributes such as day of the month, time of the day, dew point, air pressure, temperature, wind direction, wind speed, humidity, wind gust, precipitation, wind condition, and visibility.

There is no surety that all of these 12 input features are important and feasible, that is why feature selection is done. The seven features: day of the month, time of the day, temperature, air pressure, humidity, wind direction, and wind speed were selected after analyzing feature importance. The feature's rank, associated errors, and importance are displayed in Table I, and features, their range, mean value, and unit are shown in Table II.

After selecting the features, feature normalization was done to ensure a reliable and efficient wind speed prediction. Depending on the individual feature range and individual feature units the features can vastly vary from each other which inconsistently morphs the search space, thus the need for feature normalization arises to build a consistent search space. Normalization sets the feature values between 0 and 1 which makes it efficient for computation, hence reducing the computation time. The following equation (1) formalizes the normalization technique.

TABLE I: Feature importance ranking of the dataset.

Feature Rank	Feature Name	Associate Error
1	Previous wind speed	2.440
2	Temperature	2.352
3	Air pressure	2.260
4	Humidity	2.175
5	Wind direction	2.173
6	Day of the month	2.090
7	Time of the day	1.893
8	Dew point temperature	1.348
9	Wind gust	1.347
10	Precipitation	1.336
11	Visibility	1.333
12	Wind condition	1.328

$$Normal\, value, X = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Where X is the normalized value, X_i is the input value, X_{min} the minimum feature value, X_i the maximum feature value.

After feature normalization, the whole data set is split into two parts, one is a test set and another one is training. We divided the whole data set into a 90-10 ratio for the training and test sets respectively. Our proposed model is trained up with the training set and later the efficiency of our proposed scheme is compared with other existing schemes with a previously unseen test set.

IV. Performance Parameters

Various evaluation metrics and statistical testing methods are used to compare experimental simulations to validate the performance of the stated wind speed predicting model. The accompanying metrics of evaluation will be briefly displayed in the section that follows.

A. Mean Absolute Error (M.A.E.)

MAE calculates the average magnitude of errors in the forecast bypass of the direction. The mathematical expression for M.A.E. is conferred in the following equation.

TABLE II: Input features characteristics.

Input Feature	Unit	Range	Average Value
Previous wind speed	m/s	0.12-11.62	3.94
Temperature	Celsius	10.70-35.30	29.90
Air pressure	KPa	94.51-102.82	99.68
Humidity	Percentage	62-81	71.34
Wind direction	Degree	0.1-360	162.34
Day of the month	-	1-31	-
Time of the day	-	00:00-23:30	-

$$\textbf{MAE} = \frac{1}{n}\sum_{i=1}^{m}|x_{ni} - x_{pi}| \quad (2)$$

Where x_{ni} = normal value and x_{pi} = predicted value.

B. Mean Squared Error (M.S.E.)

The mean of the absolute deviations between the data set's original and forecasted values is known as the mean squared error. It calculates the variance of the residuals.

$$\textbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{d}_i - d_i)^2 \quad (3)$$

C. Root Mean Squared Error (R.M.S.E.)

If large errors are expected, then RMSE is a good approach for calculating errors. The general expression of R.M.S.E. can be expressed as,

$$\textbf{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(x_{ni} - x_{pi})^2} \quad (4)$$

Where x_{ni} = normal value and x_{pi} = predicted value.

V. Modeling Algorithms

This work applied different methods for wind speed forecasting in Newfoundland and proposed the best result for CNN, ANN, and SVM. Among the most known models used in predicting the speed of wind here, SVM, ANN and CNN are considered as they perform better than others. Before graphically analyzing, the basic concepts of SVM, ANN, and CNN are described. First, a brief introduction of the individual models, inclusive of the algorithms, and then the working process of the proposed way is narrated.

A. Proposed 1D CNN LSTM Model

Convolutional Neural Networks (CNNs) are cutting-edge deep learning techniques for image processing. The performance of CNNs, first proposed by Yann LeCun in [15], is ascribed to three fundamental elements contained in its model structure: local receptive fields, shared weights, and subsampling. CNN presented in this research is made up of convolution and pooling that are one-dimensional (1D). Unlike in a normal 2D-CNN, where kernels stride across spatial coordinates of a picture, i.e., from left to right and top to bottom, kernels in 1-dimensional CNN layers stride exclusively in one dimension, which in this case is the temporal dimension [16]. The structure is formed by a layered series, the first being convolutional layers and pooling layers. 1D CNN error function minimization is generally done by stochastic gradient descent type algorithms such as *S.G.D., Adam, Adamax, and Nadam.*

$$\boldsymbol{map}_{mn}^{uv} = rule\left(\sum_{l}\sum_{h=0}^{h'-1}\sum_{w=0}^{w'-1}\omega_{mnl}^{hw}\ .\ \boldsymbol{map}_{(m-1)l}^{(u+h)(v+w)} + b_{mn}\right) \quad (5)$$

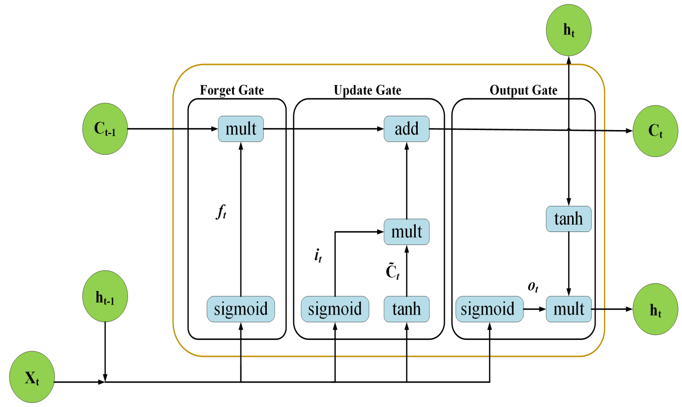

Fig. 1: Simplified LSTM architecture.

Here, The indices of such rows and columns in this case are u and v, which serves as the feature map. The convolution filter's row and column indices are, h and w, accordingly. The convolution filter's row and column counts are h' and w', respectively. The feature maps in the $(m-1)$ th layer's index is l. The bias of its n th feature map inside the m th layer is b_{mn}. ω_{mnl}^{hw} is the position value, For connecting the l th feature map in the $(m-1)$ th layer within the n th new feature map inside the m th layer (h, w) act like convolution filter, $\boldsymbol{map}_{mn}^{uv}$ is the value of such a position, (u, v) in the n th new feature map in the m th layer. $\boldsymbol{map}_{(m-1)l}^{(u+h)(v+w)}$ is the position value, The feature map in the $(m-1)$ th layer's l th feature map is $(u+h, v+w)$.

Recurrent neural network (RNN) improvement is LSTM. In overcoming the disappearing and exploding gradient problem, LSTM recommends memory blocks rather than traditional RNN units [17]. The primary distinction between LSTM and RNNs is the addition of the cell state for long-term state storage. LSTM network would remember and relate previous knowledge to current data [18]. LSTM is connected with three gates: one input gate, one output gate and one forget gate, where x_t is the current input, C_t and C_{t-1} are the current and prior cell states, accordingly, and h_t and h_{t-1} are the new and previous outputs.

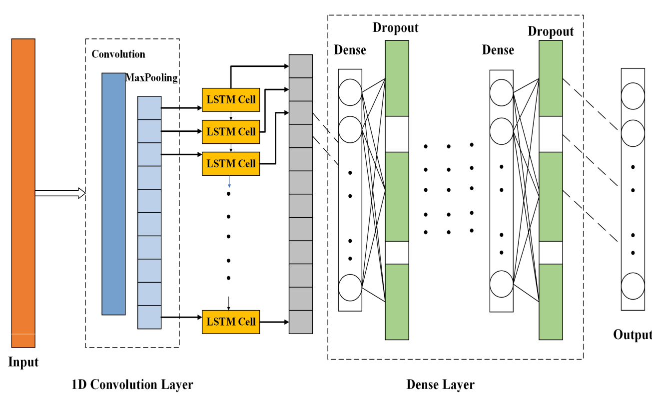

Fig. 2: Hybrid CNN-LSTM model architecture.

The input gate principle of a LSTM cell is given below:

$$i_t = \sigma \left(W_i \cdot [h_{t-1},\, x_t] + b_i \right) \tag{6}$$

$$\widetilde{C_t} = tanh \left(W_i \cdot [h_{t-1},\, x_t] + b_i \right) \tag{7}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t) \tag{8}$$

Here, equation (6) denotes that a sigmoid function is passed through h_{t-1} and x_t which determines the portion of information should be included. Afterward, equation(7) applies a $tanh$ function over h_{t-1} and x_t to gather new information. The current moment information is for storing $\widetilde{C_t}$ and the stored long term information is for storing C_t which is combined in equation (8) where i_t indicates the sigmoid output and $\widetilde{C_t}$ indicates the $tanh$ output. The trainable parameters of the LSTM cells W_i and b_i are weights matrices and bias of the input gate respectively. The forget gate of the LSTM then provides for the controlled transmission of such information via not only a sigmoid layer but also a dot product. equation (9) is used to determine whether to forget relevant information from a prior cell with a particular probability. W_f corresponds to a weight matrix, b_f is the bias, and is the sigmoid function.

$$f_t = \sigma \left(W_f \cdot [h_{t-1},\, x_t] + b_f \right) \tag{9}$$

The output gate of the LSTM determines the states necessary for continuing the h_{t-1} & x_t inputs after (12) and (13). The final output is calculated and multiplied by the state decision vectors that send fresh information, C_t, via the $tanh$ layer.

$$O_t = \sigma \left(W_o \cdot [h_{t-1},\, x_t] + b_o \right) \tag{10}$$

$$h_t = O_t * tanh(C_t) \tag{11}$$

Where weight matrices and bias of the output gate are denoted as W_o and, b_o respectively. The structural view of an LSTM cell is given in Fig. 1.

In this paper, a hybrid 1D CNN LSTM model has been developed to predict the wind speed from the previous wind speed data. The model architecture has been designed by incorporating 1D CNN with LSTM, where the CNN layer extracts the temporal feature maps of the data and the LSTM layer stores the sequence of the sequential data. The Fig. 2 depicts the proposed model for predicting wind speed. The network consists of a 1D CNN layer, a Max pooling Layer, one LSTM layer, and fully connected dense layers in conjunction with dropouts.

B. Bayesian Optimization Algorithm

Bayesian optimization algorithm (BOA) is an extension of the Bayes Theorem, which illustrates the probability of an event happening based on previous knowledge of the conditions that occur related to the event. BOA is a vigorous methodology to find the extrema of an unknown objective function. It infers the function as a sampled form of a Gaussian process and assumes it with a proxy function [19]. For the objective functions where the closed-from expression is unknown, the BOA is most effectively implemented, although several observations can be obtained from this function. In this manuscript, the BOA is used to calculate the optimum value of the model hyperparameters by minimizing the validation loss.

TABLE III: Model hyperparameter search space for Bayesian optimization.

Hyperparameter	**Search Space**
Number of convolution filters	4-512
Number of LSTM cells	4-512
Activation function	Linear, ReLU, Sigmoid, ELU, Tanh
Optimization function	Adam, SGD, Nadam, AdaDelta, Adamax, AdaGrad, RMSprop
Number of neurons in 1st hidden layer	5-512
Dropout rate	0-0.8
Batch size	4-128

$$A : \mathbb{S} \left(M_a, M_b, M_c, ..., M_n \right) \subset \mathbb{R}^n \rightarrow \mathbb{R} \tag{12}$$

Here, $\mathbb{S}$ is the search space of the hyperparameters, where the model parameters can be defined as batch size M_a, number of hidden layer M_b, dropout rate M_c etc.

To locate the specific model hyperparameter which are optimal for the required problem, the search space can be represented as $s^* \in \mathbb{S}$ such that

$$s^* = \arg\min_{s \in \mathbb{R}} A \tag{13}$$

Here, the observation's objective function is represented as $D_{1:n} = \{s_{1:n}, A(s_{1:n})\}$ that allows the BOA to construct a probabilistic model bounded in $A(s)$ enabling the manipulation of the model hyperparameters to locate the succeeding position in $\mathbb{S}$ to sample.

By determining the posterior distribution of the objective function utilizing Bayes' theorem, BOA selects the next set of hyperparameters within this set of distributions. BOA calculates the output of the previous sample point of the distribution to pick up the shape of the objective function. This information in terms of sample output allows the BOA to detect the set of hyperparameters that maximizes the target output.

In our manuscript, the target output is the negative of the validation loss function that needs to be maximized in order to find the optimal hyperparameters which are produced by the objective function. Hyperparameter search space for Bayesian optimization is illustrated in Table III.

VI. Result and analysis

The final result is divided into two sections. Firstly, the detail of model performance is illustrated, and later the proposed hybrid model is compared with our established ML and DL models. The experiments attained in this paper were

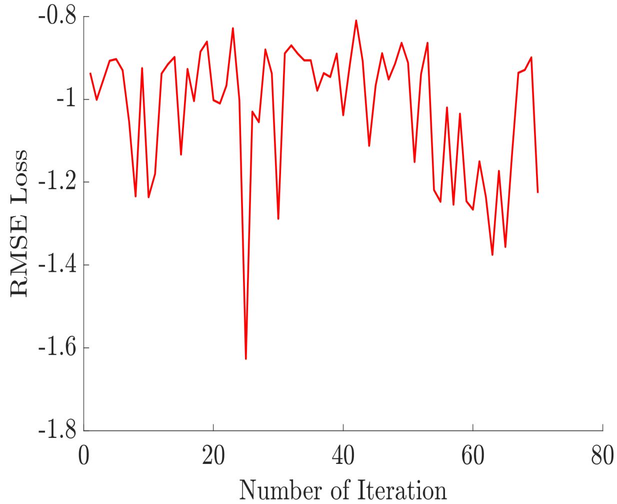

Fig. 3: Bayesian Optimization iterations of the proposed model.

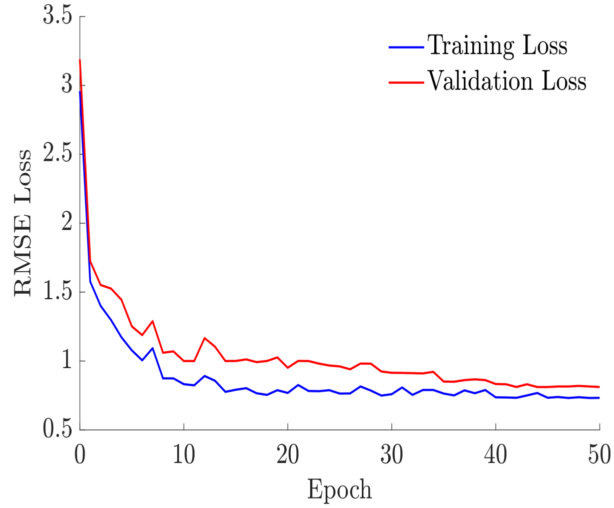

Fig. 4: Change in R.M.S.E. with number of epochs of the proposed model.

executed utilizing Keras, Scikit-learn, and TensorFlow libraries which were written in Python on an Intel Core i5-7300HQ CPU running at 2.50 GHz, 8 GB RAM, and a x64-based processor.

A. Model performance

The optimization result of the model hyperparameters is shown within the Fig. 3 graph. These hyperparameters were selected by Bayesian optimization algorithm, which performed sovereign results without any expert tempering. The optimum configurations of the model are described in Table IV. The Bayesian optimization function ran for 70 calls to locate the best combination of hyperparameters. Where the optimum result for CNN-LSTM model was found at RMSE loss of 0.81 on the 42nd iteration.

CNN-LSTM models were evaluated for 60 epochs with the hyperparameter functions and values obtained from the Bayesian optimization. Fig. 3 illustrates the training and validation loss curve of the proposed model, where the RMSE loss reduces over time, which implies that the model was learning from the training data-set. The figures show that after a few epochs, the model adapted to learn how to predict wind speeds using given input factors. The model loss fluctuated over certain epochs, attributing to the dropout rate within the hidden layer. Additionally, the small distance between training and validation curves signifies that the models had less over-fitting and hidden layer dropout decreases over fitting and improves computational performance. In Fig. 4 the predicted wind speed of our proposed model is compared with ground truth , where it is demonstrated that the proposed model is able to learn the trend of wind speed that implies that these models are efficient to predict future wind speed from previous data.

TABLE IV: Selected hyperparameter values of Bayesian optimization.

Hyper-paramters	**Selected Hyper-parameter Values**
No. of convolution filters	350
No. of LSTM cells	326
Activation function	Relu
Optimization method	Adam
No. of neuron in hidden layer	488
Dropout rate	0.28
Batch size	39

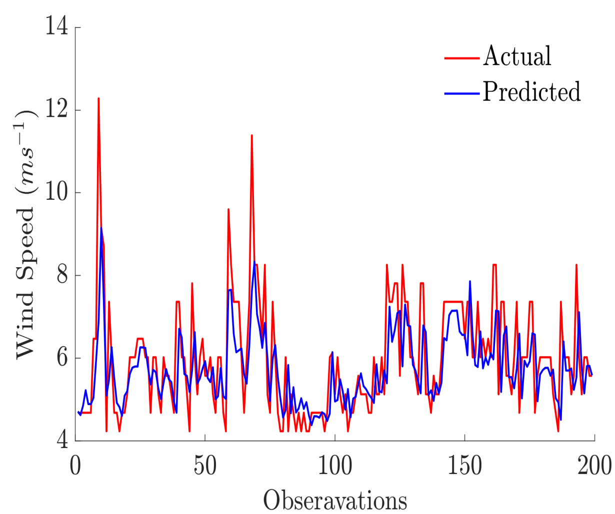

Fig. 5: Predicted wind speed by the proposed model for a section of test samples.

TABLE V: Performance comparison of different models

Model	MAE	MSE	RMSE
SVM	1.04	1.85	1.36
ANN	0.98	1.71	1.31
CNN	0.63	1.00	1.00
LSTM	0.61	0.98	0.99
Proposed CNN-LSTM	**0.48**	**0.81**	**0.66**

B. Comparison

Other robust and well suited models such as, SVM, ANN, CNN and LSTM were also conducted to predict wind speed with the same data set, where it can be seen that the performance of ANN and SVM models are workable to predict wind speed, but they could not overcome the other three models in terms of performance. MAE, RMSE and MSE of all these five models are presented in Table V. From Table V, it is justified that CNN, LSTM and CNN-LSTM models surpass ANN and SVM models. Though, the performance of these three models are efficient and feasible in terms of all of these three evaluation metrics. MAE, RMSE, MSE value for the proposed model is correspondents to 0.48, 0.81 and 0.66. So, in terms of the performance metrics, our proposed Bayesian optimized CNN-LSTM model presents the leading performance and surpass the other established models.

VII. Conclusion

Wind speed forecasting engages an exigent role in the field of renewable energy systems and management. This paper introduces a hybrid model using CNN and LSTM, where Bayesian optimization is used to pick up the best hyperparameters values to find out the optimal performance to predict wind speed. A powerful and efficient model was required to define the non-linear relation between wind speed and different input features, which is why the CNN model is used in this study to extract the input features and LSTM to predict future wind speed. MAE, RMSE, and MSE are three useful and prominent performance metrics used in this paper for performance evaluation. In this paper, our proposed network is compared with other four powerful and established algorithms such as SVM, ANN, CNN, and LSTM. Though ANN and SVM models have comparatively higher MAE, RMSE, and MSE, the other three DL models CNN, LSTM, and proposed CNN-LSTM perform effectively. our proposed model surpasses the other in terms of all three performance metrics where MAE, RMSE, and MSE. The performance metrics values indicate that our model is efficient to identify the hidden feature and predict accurately.

References

[1] T. Niet, N. Arianpoo, K. Kuling, and A. S. Wright, "Embedding the united nations sustainable development goals into energy systems analysis: expanding the food–energy–water nexus," *Energy, Sustainability and Society*, vol. 11, no. 1, pp. 1–12, 2021.

[2] A. Soroudi and M. Ehsan, "A possibilistic–probabilistic tool for evaluating the impact of stochastic renewable and controllable power generation on energy losses in distribution networks—a case study," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 794–800, 2011.

[3] H. E. Murdock, D. Gibb, T. André, J. L. Sawin, A. Brown, L. Ranalder, U. Collier, C. Dent, B. Epp, C. Hareesh Kumar *et al.*, "Renewables 2021-global status report," 2021.

[4] A. Clerc, M. Anderson, P. Stuart, and G. Habenicht, "A systematic method for quantifying wind flow modelling uncertainty in wind resource assessment," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 111, pp. 85–94, 2012.

[5] "Bangladesh meteorological department," https://live3.bmd.gov.bd/, accessed: 2022-11-5.

[6] A. Troncoso, S. Salcedo-Sanz, C. Casanova-Mateo, J. Riquelme, and L. Prieto, "Local models-based regression trees for very short-term wind speed prediction," *Renewable Energy*, vol. 81, pp. 589–598, 2015.

[7] A. Lahouar and J. B. H. Slama, "Hour-ahead wind power forecast based on random forests," *Renewable energy*, vol. 109, pp. 529–541, 2017.

[8] J. Zhou, J. Shi, and G. Li, "Fine tuning support vector machines for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 52, no. 4, pp. 1990–1998, 2011.

[9] J. L. Torres, A. Garcia, M. De Blas, and A. De Francisco, "Forecast of hourly average wind speed with arma models in navarre (spain)," *Solar energy*, vol. 79, no. 1, pp. 65–77, 2005.

[10] X. Wang, P. Guo, and X. Huang, "A review of wind power forecasting models," *Energy procedia*, vol. 12, pp. 770–778, 2011.

[11] H. Rahaman, T. R. Bashar, M. Munem, M. H. H. Hasib, H. Mahmud, and A. N. Alif, "Bayesian optimization based ann model for short term wind speed forecasting in newfoundland, canada," in *2020 IEEE Electric Power and Energy Conference (EPEC)*. IEEE, 2020, pp. 1–5.

[12] X. Zhu, R. Liu, Y. Chen, X. Gao, Y. Wang, and Z. Xu, "Wind speed behaviors feather analysis and its utilization on wind speed prediction using 3d-cnn," *Energy*, vol. 236, p. 121523, 2021.

[13] Y. Yan, X. Wang, F. Ren, Z. Shao, and C. Tian, "Wind speed prediction using a hybrid model of eemd and lstm considering seasonal features," *Energy Reports*, vol. 8, pp. 8965–8980, 2022.

[14] Z. Shen, X. Fan, L. Zhang, and H. Yu, "Wind speed prediction of unmanned sailboat based on cnn and lstm hybrid neural network," *Ocean Engineering*, vol. 254, p. 111352, 2022.

[15] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. IEEE, 2012, pp. 3288–3291.

[16] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, and D. J. Inman, "Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks," *Journal of Sound and Vibration*, vol. 424, pp. 158–172, 2018.

[17] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[18] G. Chen, "A gentle tutorial of recurrent neural network with error backpropagation. 2016."

[19] M. Aoki, "On some convergence questions in bayesian optimization problems," *IEEE Transactions on Automatic Control*, vol. 10, no. 2, pp. 180–182, 1965.