

Original Article

머신 러닝을 활용한 단기 시계열 풍속 예측

김도영[†]

홍익대학교 조선해양공학과 교수

Short Term Time Series Prediction on Wind Speed using Machine Learning

Do Young Kim[†]

Professor, Department of Naval Architecture and Ocean Engineering, Hongik University, Sejong 30016, Korea

요 약

소형 풍력 터빈이 안전하게 작동하고 갑작스러운 돌풍으로 인한 손상을 방지하려면 가까운 미래의 풍속의 단기 예측이 필요하다. 본 연구에서는 풍력 터빈 현장에서 얻은 풍속 데이터를 사용하여 시계열에서 풍속을 예측하기 위해 여러 머신 러닝 모델을 이용을 하여 비교를 한다. 분석에 사용된 풍속 데이터는 1Hz로 재샘플링하였다. 본 연구에서는 세 개의 인공 신경망을 사용하여 시계열 예측을 수행하였다. 세 가지 모델은 순환 신경망(RNN), 장단기 메모리(LSTM), 그리고 게이트 순환 유닛(GRU)이다. 모든 모델은 TensorFlow와 Keras를 사용하여 구현하였다. 모델을 학습하기 위해 배치(batch) 크기는 32, 500개의 에포크(epoch)가 사용하였고 Adam 옵티마이저(optimizer)를 사용하였다. 손실(loss)은 평균 제곱근 오차와 평균 절대 오차를 사용하였고 20%의 드롭아웃(dropout)이 적용되었다. 풍속 데이터는 120초 간격으로 구분되며, 처음 110초는 훈련 데이터로 사용되고 마지막 10초는 테스트 데이터로 사용하였다. 모든 모델이 단기 시계열 풍속예측이 가능하였으며, 세 가지 모델 중 RNN 모델은 짧은 시간 내에 풍속이 빠르게 변화하는 경우 다른 모델보다 풍속을 상대적으로 적은 오차로 더 잘 예측할 수 있었다.

Abstract – In order for small wind turbines to operate safely and to prevent possible damage due to sudden gust, one needs short time prediction of wind velocity in the near future. In this research, we compare several machine learning models to forecast wind speed using wind speed data obtained at wind turbine site. The wind data that is used for the analysis is resampled at 1 Hz. In this study we used three artificial neural networks in making a time series forecast. Three models are recurrent neural network (RNN), long sort-term memory (LSTM), and gated recurrent unit (GRU). All models are implemented using TensorFlow and Keras. To train models 500 epochs are used with batch size 32 for most of the cases. Adams optimizer is used. And the loss is based on root mean square error and mean absolute error. 20% dropout is applied. Wind data are segmented into 120 seconds intervals and the first 110 seconds are used for training data and final 10 seconds are used for test data. All three models can forecast short-term wind speeds. Among them, RNN model can predict wind speed relatively better than other models when the wind speed is changing rapidly in a short duration.

Keywords: Wind speed forecasting(풍속예측), Machine learning(기계학습), Artificial neural network(인공신경망), Times Series forecasting(시계열예측), Recurrent neural network(순환신경망)

1. Introduction

[†]Corresponding author: dykim@hongik.ac.kr
This paper is expanded and revised version based on conference paper in TEAM 2023, October 10–13, 2023, Busan, South Korea.
(이 논문은 TEAM 2023, October 10–13, 2023, Busan, South Korea. 발표된 논문을 확장·개정한 버전이다)

Renewable energy is derived from natural sources, and it is becoming an important part of energy policy worldwide to protect our environment and alleviate climate change. Among various renewable energy sources, wind turbines play a significant

role in renewable energy programs. However, wind turbines are subject to variations in wind speed. Wind turbines are susceptible to possible damage on the wind turbines structures and grid system due to sudden increase of wind speed. For large scale turbines, controller on nacelle can be used to make control depending on the wind speed. But this is not economically feasible for small wind turbines.

Short-term wind speed forecasting is therefore critical for improving operational safety and reliability. Very short-term and short-term predictions ranging from seconds to several minutes are especially important for turbine control, real-time grid operation, and damage prevention, as forecasting uncertainty grows significantly with increasing time horizons (Hanifi *et al.*, [2020]).

So, an accurate prediction of wind speed is significant for the operation of power system safely and efficiently (Hong *et al.* [2018]; GWEC [2019]; Yang *et al.* [2019]; Zhang *et al.* [2019]). In this research, we aim to make short-term prediction on wind speed using wind speed data obtained at wind turbine site. Predicted wind speed can be used for small wind turbines to operate safely and prevent possible damage due to sudden gust. Machine learning techniques are used to predict waves and wind speed in the oceans (Kim [2020]).

Wind forecasting methods are commonly classified into physical, statistical, and hybrid approaches. Physical models rely on numerical weather prediction (NWP) and detailed atmospheric modeling to estimate wind conditions at turbine hub height. While these methods can achieve good accuracy for medium- and long-term forecasting, they are computationally intensive and less suitable for real-time short-term applications (Lange & Focken [2006]). Statistical and machine learning-based approaches, in contrast, use historical measurements to learn patterns in wind behavior and are widely favored for short-term prediction

due to their lower computational cost and adaptability (Zhao *et al.* [2011]; Hanifi *et al.* [2020]).

Among statistical approaches, artificial neural networks have received substantial attention for wind speed and wind power forecasting. These models are capable of capturing nonlinear relationships between input features and target variables without requiring explicit physical modeling. Recurrent neural network-based architectures, including recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and gated recurrent unit (GRU) networks, are particularly effective for time-series forecasting because they can model temporal dependencies and sequential correlations in wind data (De Giorgi *et al.* [2011]; Zhang *et al.* [2019]). Prior studies have shown that such models often outperform conventional time-series techniques, especially under rapidly changing wind conditions (Hanifi *et al.* [2020]).

Motivated by these findings, this study focuses on short-term wind speed prediction using high-frequency wind data collected at a wind turbine site. Three recurrent neural network-based models—RNN, LSTM, and GRU—are implemented and evaluated for their forecasting performance. By systematically comparing their predictive accuracy during periods of rapid wind variation, this research aims to identify an effective and computationally efficient approach for short-term wind speed forecasting that can enhance the safe operation of small wind turbines.

2. Wind Data

The wind data was measured at the wind turbine sites. Each data file stores wind data for 24 hours for a specific date. Wind data contained wind speed, wind direction, temperature, humidity and air pressure and were sampled at 1 Hz. Among the data set, we pick up wind data on 2023-09-23. The maximum wind

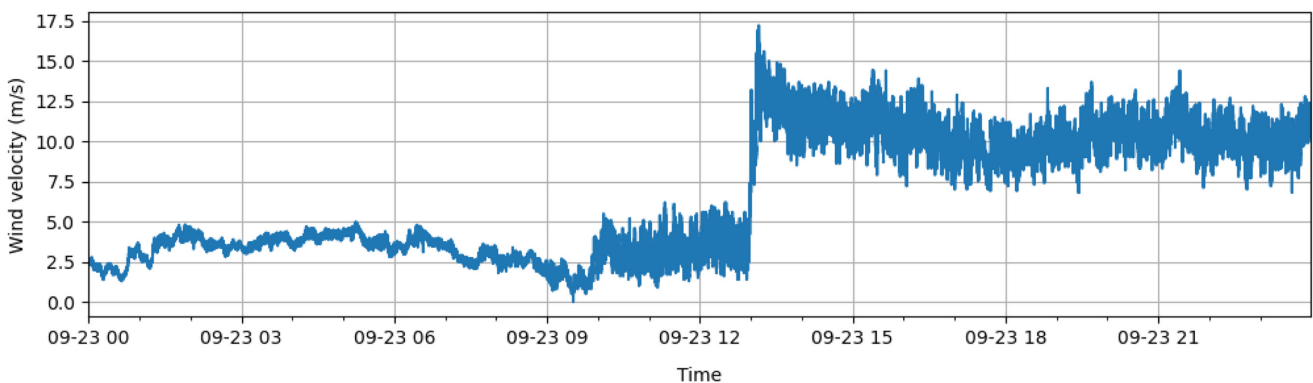


Fig. 1. Wind speed.

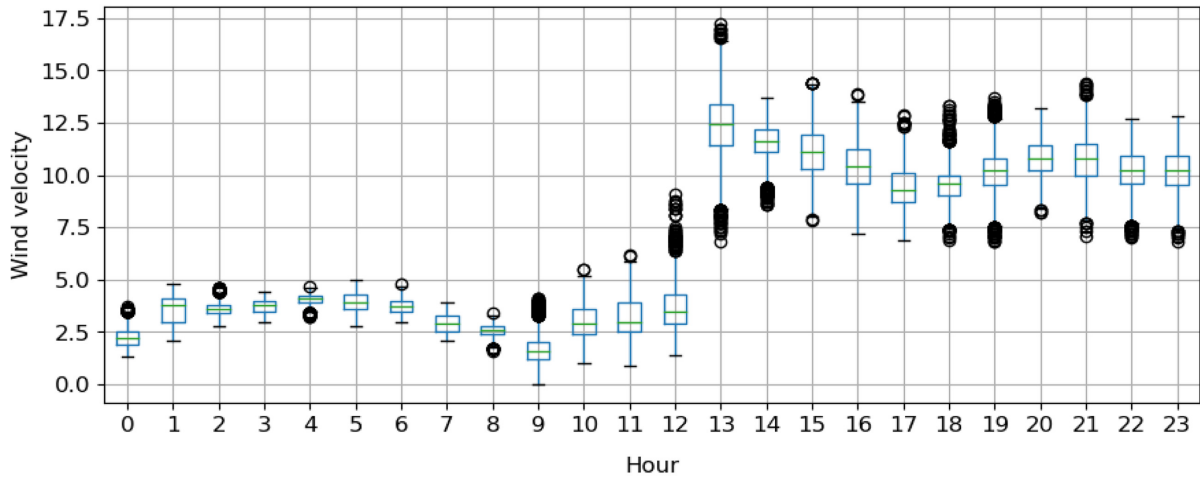


Fig. 2. Boxplots of wind speed grouped by hour.

speed in this data is 17.2 m/s. The wind speed changes rapidly from average of 2.5 m/s to over 15 m/s within 20 minutes. There are 86,400 data lines and there are no null data. Figure 1 shows the wind velocity on September 23, 2022. Wind speed does not change much until 09:00, but wind speed starts to fluctuate as of 09:30. Wind speed rapidly increases after 13:00 and shows high wind speeds and high variability. In this region, we need to control pitch of the blades so that the power generated are maintained under a certain level to protect the wind turbine system.

Figure 2 shows the box plots of wind speed grouped by hour. Until 8:00 the average wind speed is under 5.0 m/s, the IQR (inter quartile range) is less than 1, and there are not many outliers. For time range 9:00~12:00, average wind speed is less than 5 m/s, but IQR becomes higher, and many outliers are appearing. Around 13:00, maximum wind speed occurred, and the average wind speed is over 10 m/s and IQR is about 2.3 m/s.

Wind data may be categorized as following four groups. 1) Wind speed is less than 5 m/s. No power control is needed; 2) Wind speed is less than 5 m/s. But variation of wind speed starts increasing; 3) Wind speed rapidly increases and reaches above 10 m/s in 10 to 15 minutes. Control of power is needed.; 4) Average wind speed is over 10 m/s, so power control is needed continuously.

3. Data Preparation

For the time series forecasting, we chose two regions from the four groups mentioned in the previous sections. Figure 3 and 4 show wind speed of the two regions. Time span of region 1 is from 12:57 to 13:02, where wind speed changes rapidly and reaches above 10 m/s. In this region, the highest instantaneous rate of wind speed change occurs. Time span of the regions 2 is from 13:05 to 13:10. In this region, the average wind speed is over 10 m/s, and the highest wind speed occurs.

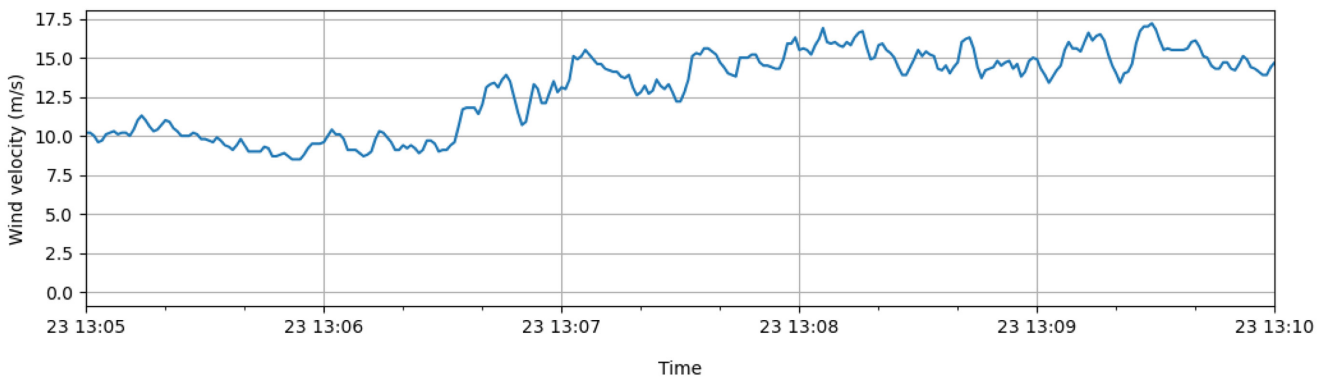


Fig. 3. Wind speed in region 1 (12:57~13:02).

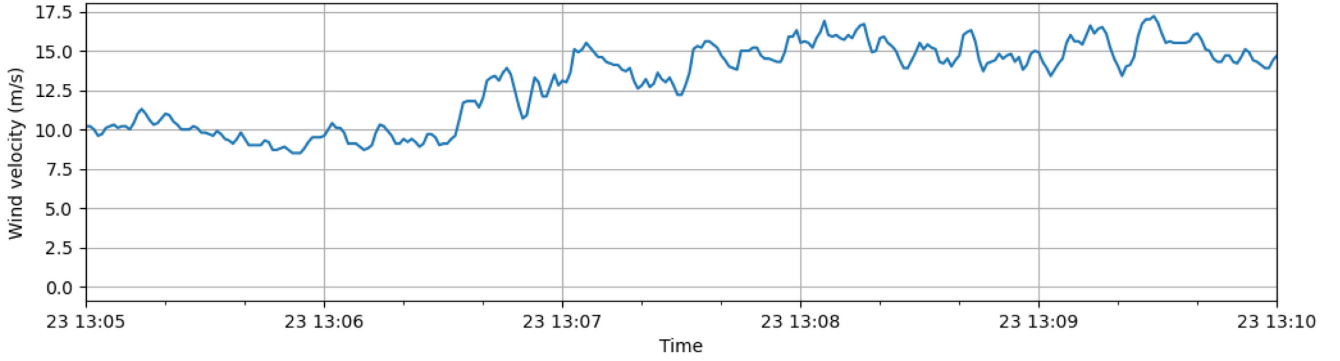


Fig. 4. Wind speed in region 2 (13:05~13:10).

4. Forecasting Models

Neural Networks automatically can learn input data's complex mappings and provide inputs and outputs. Among artificial neural networks, RNN (Recurrent Neural Network) uses hidden layer's output based on the previous input. Based on these characteristics, RNN is used for language and time-series data processing (Simplilearn [2025]). LSTM (long short-term memory) is one type of RNN that has the characteristic of being able to extract patterns of sequences that were processed over a long time (MATLAB [2025]). GRU is also a type of RNN that is abbreviated as Gated Recurrent Unit. GRU has fewer parameters than LSTM due to having no output gate, but it shows similar characteristics with LSTM in terms of speech recognition (Techopedia [2025]). These models can provide decent time-series forecasting with proper change and update of information that is processed with the time-series data.

RNN is a neural network that uses memory from the hidden state to update the memory. RNN has an internal hidden state (h_t) at the current time t . Previous hidden state (h_{t-1}) contains previous inputs that allow to process information of the current input (x_t) into hidden state. Information of the current and previous inputs then uses tanh activation from previous hidden state to the internal hidden state. RNN also uses the gradient descent that allows to determine weight (W) values in order to calculate and minimize the loss values [9].

$$h_t = \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh}) \quad (1)$$

where

- h_t : hidden state at the current time t
- h_{t-1} : previous hidden state at the current time $t-1$
- x_t : input at time t
- W : Weight
- b : bias

LSTM is a specific type of RNN that has a strong advantage of long-term prediction and is also widely used for time-series data. Like RNN, LSTM also has the cell state (C_t) that allows to process previous information that is sent from previous state. This type of RNN can process addition and multiplication of the information that needs to be entered and deleted from the input gates. In order to maintain and process necessary information, LSTM has three types of gates. First is input gate (i_t) which makes the information to process from the previous cell state. Second is forget gate (f_t) which allows to determine which information that can be forgotten. Third is output gate (h_t, o_t), which allows to decide where the next information should be. LSTM uses two types of function (tanh activation and sigmoid function) in order to determine which information should be in each cell state for each gate. Sigmoid function allows the input values to be close to either 0 or 1 (Lazzerii [2020]). Due to these attributes, LSTM has a benefit of not leaving critical information that is needed to pass just in case the gradient value is too small.

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (2)$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (3)$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (4)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (5)$$

$$c_t = f_t c_{t-1} + i_t * g_t \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

where

- i_t : input gate
- f_t : forget gate
- c_t : cell state
- h_t, o_t : output gate
- σ : sigmoid function

GRU is also a type of RNN, but GRU has a simpler structure and design than LSTM. For GRU, it just uses the hidden state to transfer information without the cell state. GRU also has three types of gates. Update gate (z_t) determines how much past information has to be passed. Second, reset gate (r_t) decides how much past information should be forgotten. Third, current gate allows to reset without a big impact from past information (Lazzerii [2020]). Due to simpler structure, GRU has a faster and efficient process than LSTM.

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \quad (8)$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \quad (9)$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t(W_{hn}h_{t-1} + b_{hn})) \quad (10)$$

$$h_t = (1 - z_t) * n_t * z_t * h_{t-1} \quad (11)$$

where

- h_t : new hidden state
- h_{t-1} : previous hidden state
- r_t : reset gate
- z_t : update gate
- σ : sigmoid function

5. Construction of Recurrent Neural Networks

The recurrent neural network models are constructed using TensorFlow and Keras and contain following process.

- Define sequential class
- Split features and targets
- Split train and test data

Table 1. Parameters setups

Parameters	Settings
Epoches	500
Batch size	32
Dense	1
Optimizer	adam
Loss:	Root Mean Squared Error (), Mean Absolute Error ()
Dropout rate	0.2

Table 2. Output shapes and parameter numbers of each layer

Layer	Output shape	Parameter Numbers
RNN	(None, 128)	16,640
LSTM	(None, 128)	66,560
GRU	(None, 128)	50,304
Drop out	(None, 128)	0
Dense	(None, 1)	129

- Create models
- Training models
- Validation of models

For an optimizer Adam optimizer is used. Two evaluation metrics is used for the loss calculation. They are root mean square error (RMSE) and mean absolute error (MAE). We are adding a dropout layer which has 20% dropout rate. Table 1 shows setups of each model. Table 2 shows output shapes and parameter numbers to be trained in each layer. Each model has following setups.

6. Results

In each region, 120-seconds data segmented are selected and they are split into the training and test data. Total of 25 simulations are made for this case. The data set is normalized using standardization. After computation, the results are inversely transformed to the original scale for presentation. The first 110 seconds are used for training and the last 10 seconds are used for testing. Among 25 simulations, 8 cases are presented in Figure 5~Figure 12. In each figure, the training data and the fitted results are shown for 40 seconds before forecasting and the actual and forecasting results for 10 seconds are shown.

In all cases, three artificial recurrent neural networks can be fitted into the training data well. Forecasting results generally

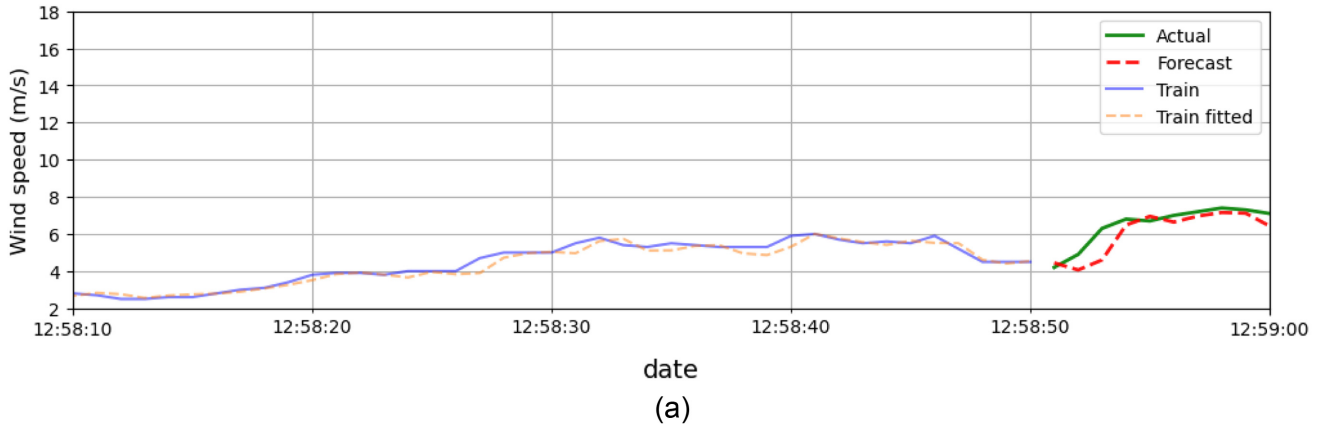
Table 3. RMSE of Test Evaluation

	RNN	LSTM	GRU
Case 1	0.6846	1.0658	0.7001
Case 2	0.9662	0.9866	0.9767
Case 3	0.3734	0.9601	0.6460
Case 4	0.6452	1.5352	1.2577
Case 5	0.5838	1.6955	0.5940
Case 6	0.2440	0.2477	0.3153
Case 7	0.6785	1.1010	0.9780
Case 8	0.8278	1.2322	1.0224

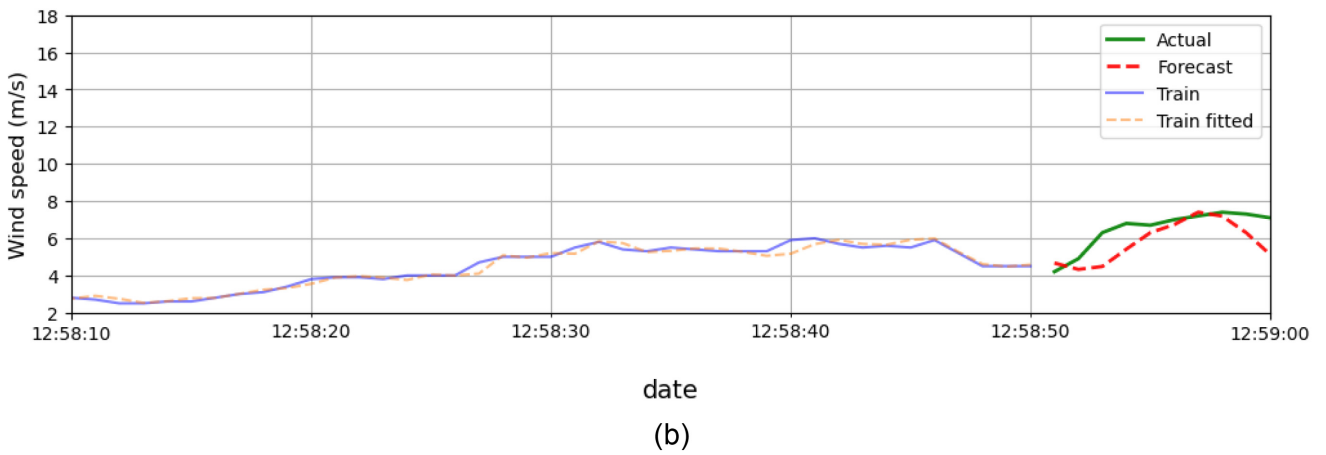
Table 4. MAE of Test Evaluation

	RNN	LSTM	GRU
Case 1	0.5132	0.8456	0.5402
Case 2	0.8744	0.8880	0.8378
Case 3	0.2999	0.8500	0.6027
Case 4	0.5413	1.0884	1.0263
Case 5	0.5263	1.5969	0.5301
Case 6	0.1836	0.1468	0.2737
Case 7	0.5527	0.8921	0.7663
Case 8	0.6742	1.0648	0.7888

Model: RNN (train 2), RMSE=0.6846, MAE=0.5132



Model: LSTM (train 2), RMSE=1.0658, MAE=0.8435



Model: GRU (train 2), RMSE=0.7001, MAE=0.5402

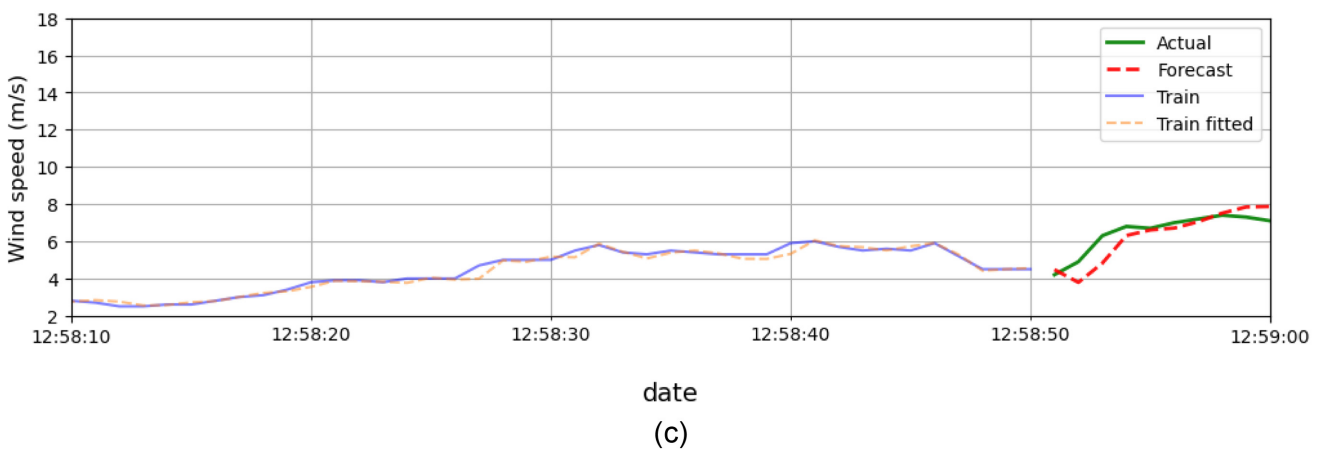
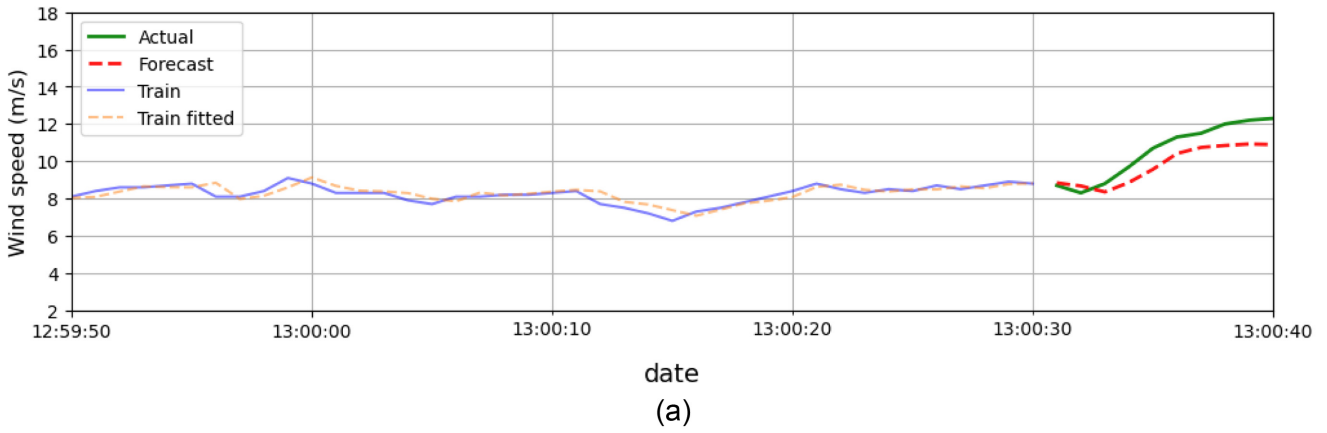


Fig. 5. (a) Case 1. Test Evaluation (Forecast at 12:58:50), (b) Case 1. Test Evaluation (Forecast at 12:58:50), (c) Case 1. Test Evaluation (Forecast at 12:58:50).

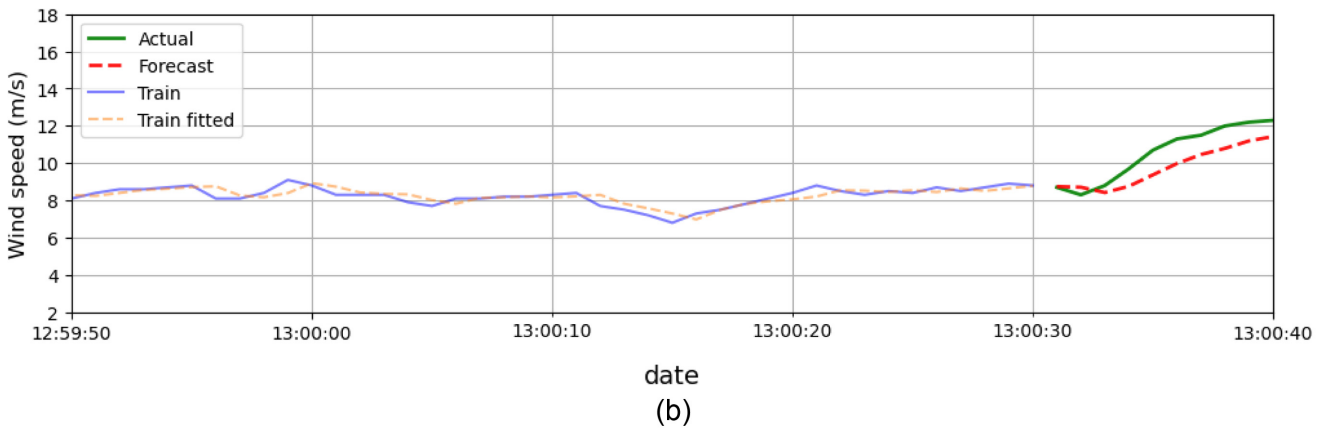
follow the changing pattern of the wind speed. In overall, RNN model gives the good estimate of wind velocity. LSTM models

sometimes overpredict (Case 3) or underpredict (Case 5) the results.

Model: RNN (train 2), RMSE=0.9662, MAE=0.8744



Model: LSTM (train 2), RMSE=0.9866, MAE=0.888



Model: GRU (train 2), RMSE=0.9767, MAE=0.8378

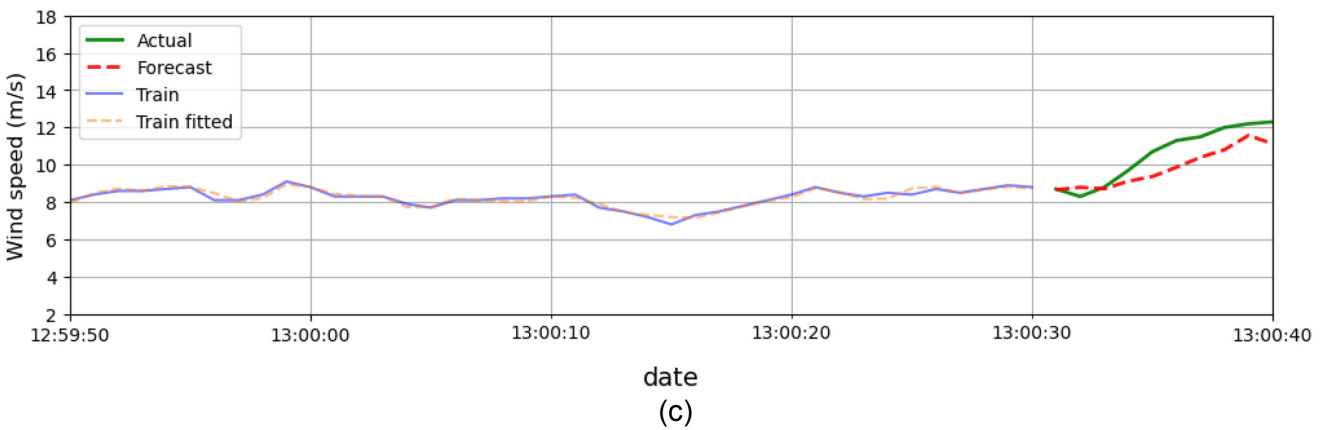
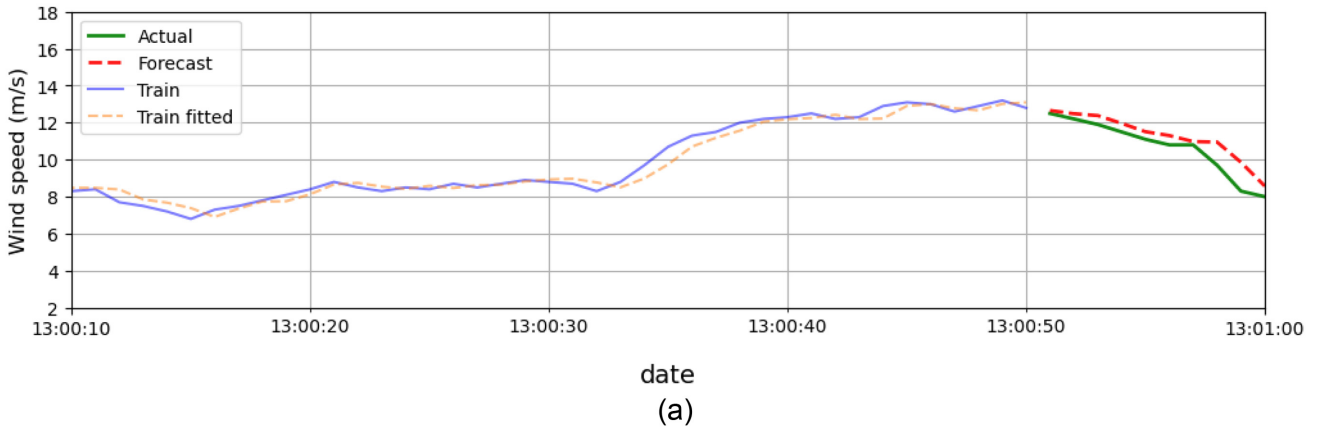


Fig. 6. (a) Case 2. Test Evaluation (Forecast at 13:00:30), (b) Case 2. Test Evaluation (Forecast at 13:00:30), (c) Case 2. Test Evaluation (Forecast at 13:00:30).

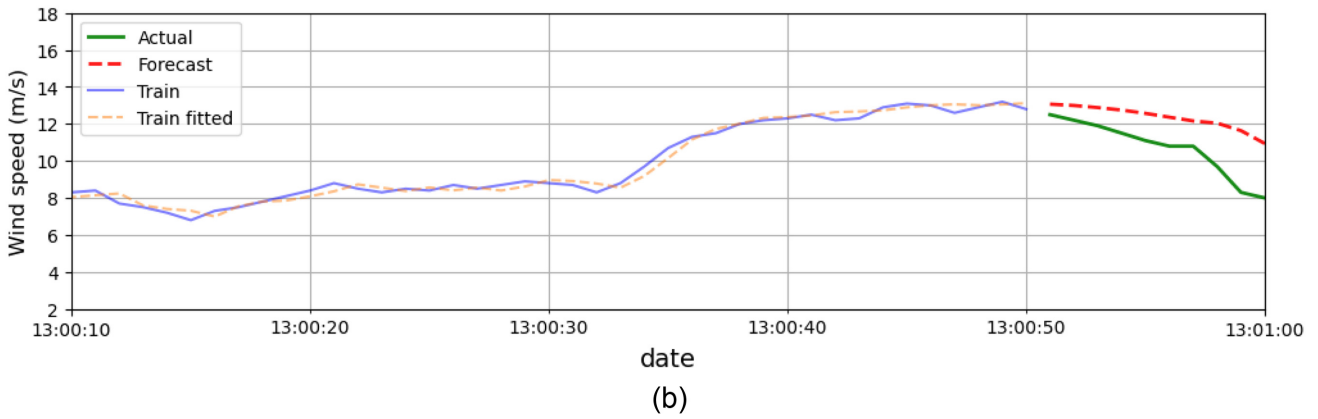
Table 3 compares root mean square errors (RMSE) of the test evaluations. For RNN, RMSE results are less than 1.0 m/s,

and the highest error occurs in case 2 where the wind speed changes most rapidly. But the RNN error is less than errors of

Model: RNN (train 2), RMSE=0.3734, MAE=0.2999



Model: LSTM (train 2), RMSE=0.9601, MAE=0.85



Model: GRU (train 2), RMSE=0.646, MAE=0.6027

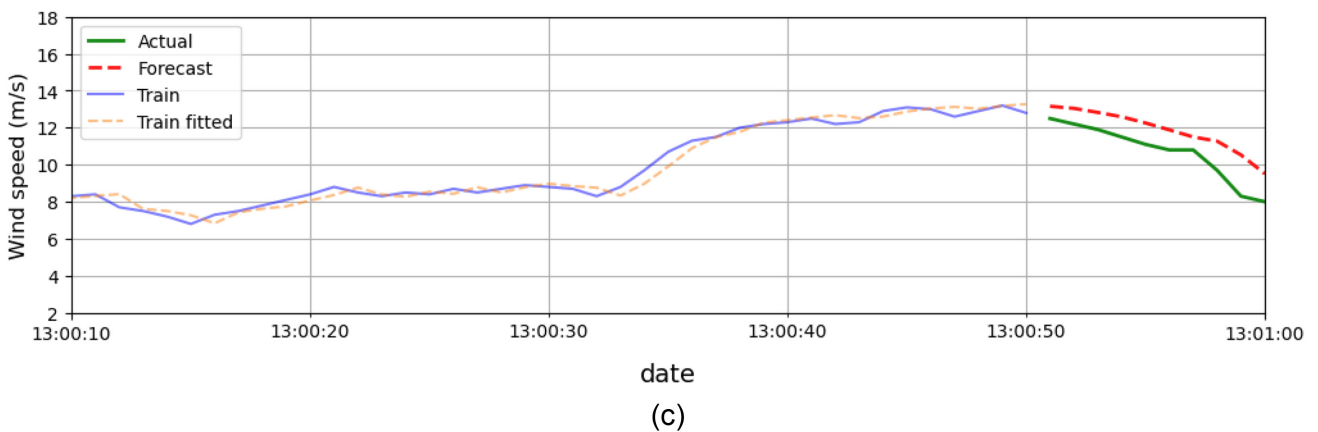
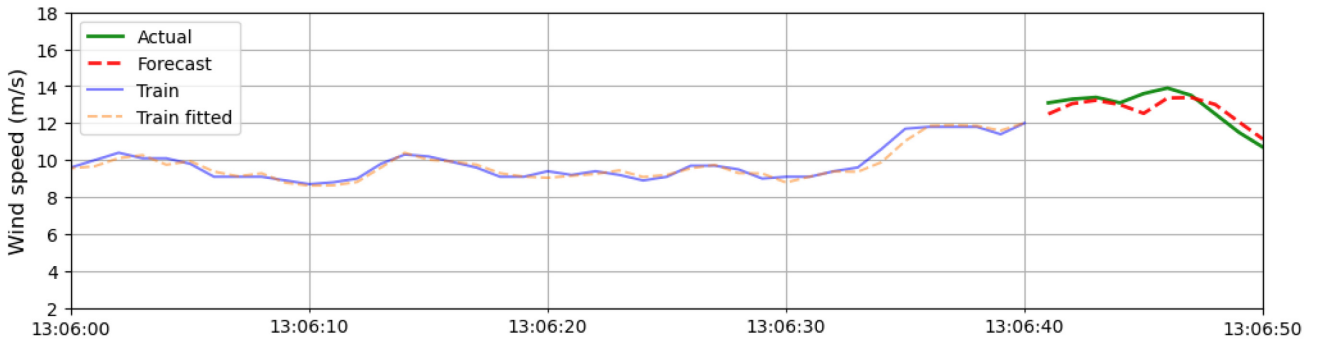


Fig. 7. (a) Case 3. Test Evaluation (Forecast at 13:00:50), (b) Case 3. Test Evaluation (Forecast at 13:00:50), (c) Case 3. Test Evaluation (Forecast at 13:00:50).

LSTM or GRU. For LSTM, the highest error occurs in case 5. For GRU, case 4 is the highest. Table 4 compares mean absolute

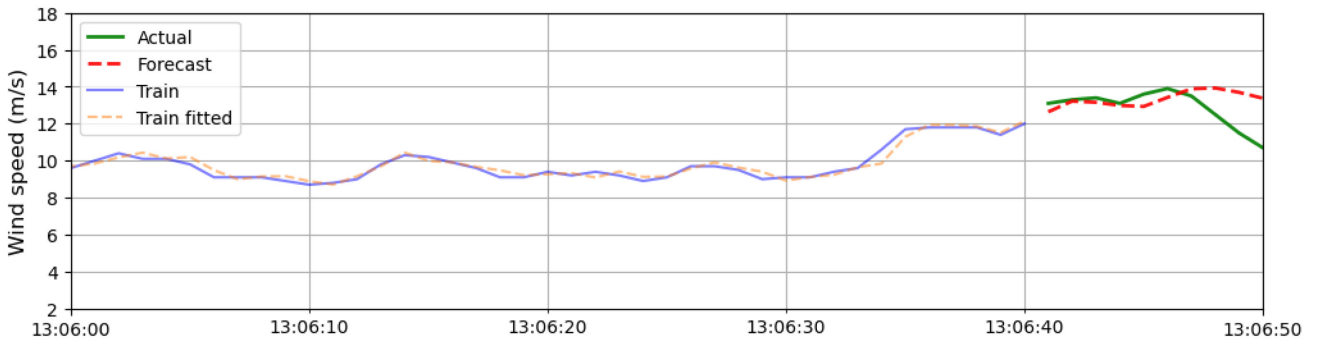
errors (MAE). For RNN, MAE results are less than 0.9 m/s, and the highest error occurs in case 2. In case 2, GRU gives slightly

Model: RNN (train 2), RMSE=0.6452, MAE=0.5413



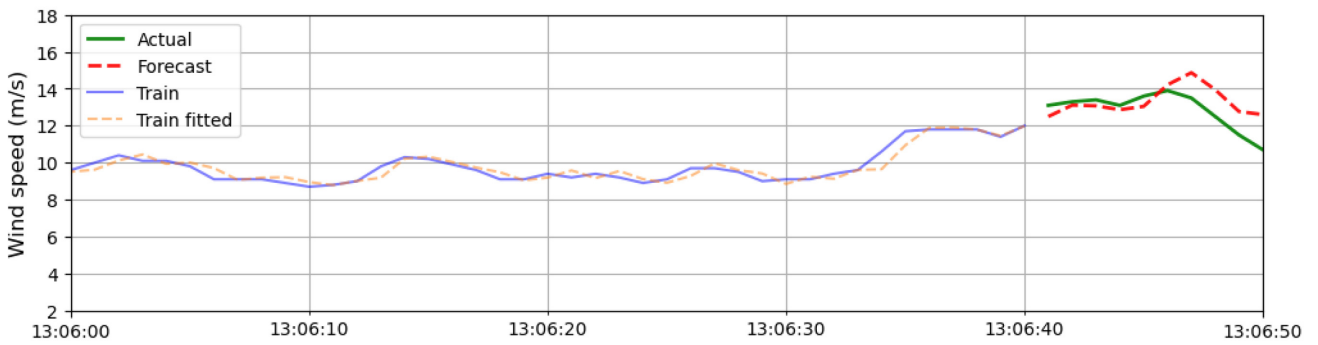
(a)

Model: LSTM (train 2), RMSE=1.5352, MAE=1.0884



(b)

Model: GRU (train 2), RMSE=1.2577, MAE=1.0236



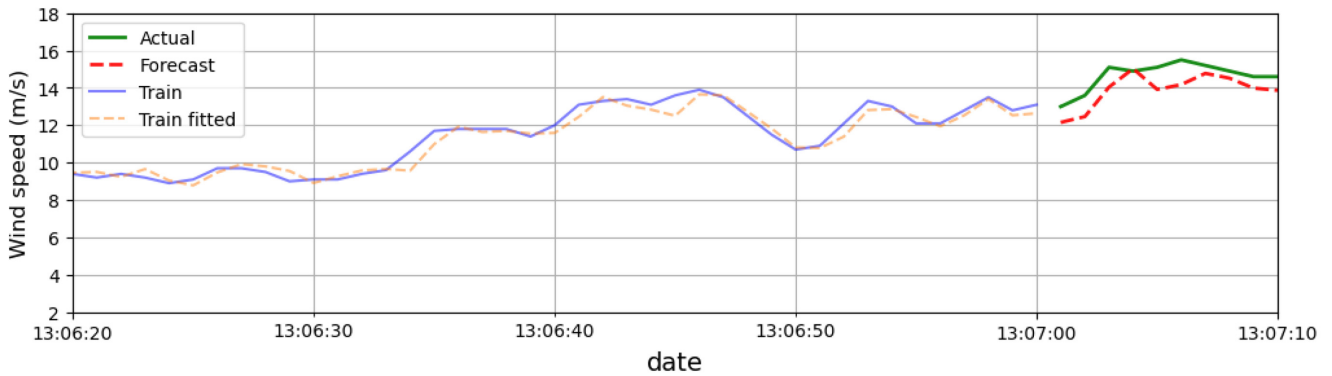
(c)

Fig. 8. (a) Case 4, Test Evaluation (Forecast at 13:06:40), (b) Case 4, Test Evaluation (Forecast at 13:06:40), (c) Case 4, Test Evaluation (Forecast at 13:06:40).

less MAE. Generally, results of RSME and MAE of RNN are less than results of other models. Based on this error estimate,

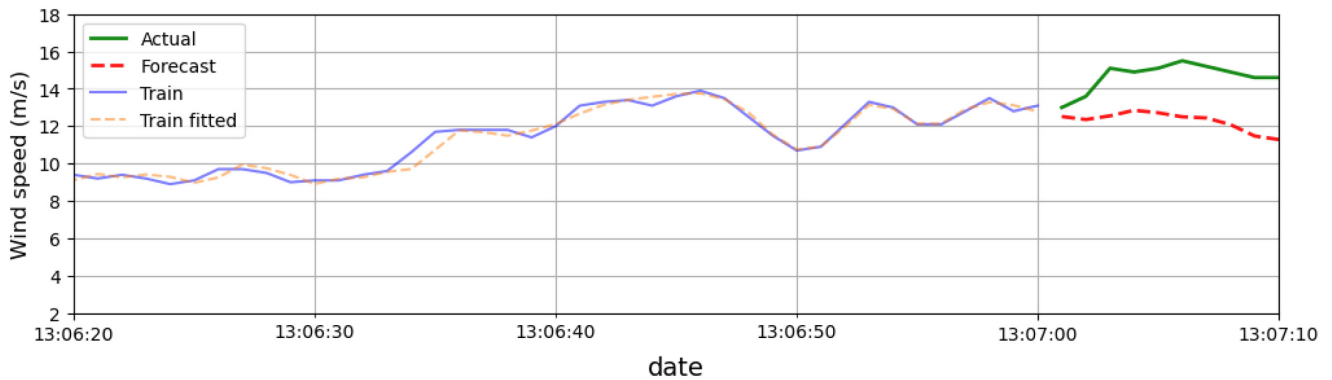
average wind speed prediction of RNN can be made within error bound less than 1.0 m/s.

Model: RNN (train 2), RMSE=0.5838, MAE=0.5263



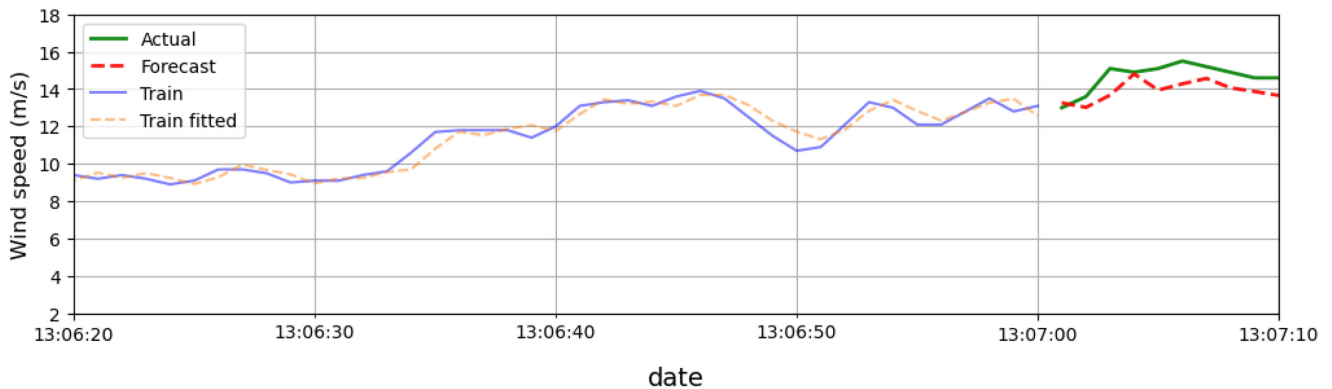
(a)

Model: LSTM (train 2), RMSE=1.6955, MAE=1.5969



(b)

Model: GRU (train 2), RMSE=0.594, MAE=0.5301



(c)

Fig. 9. (a) Case 5. Test Evaluation (Forecast at 13:07:00), (b) Case 5. Test Evaluation (Forecast at 13:07:00), (c) Case 5. Test Evaluation (Forecast at 13:07:00).

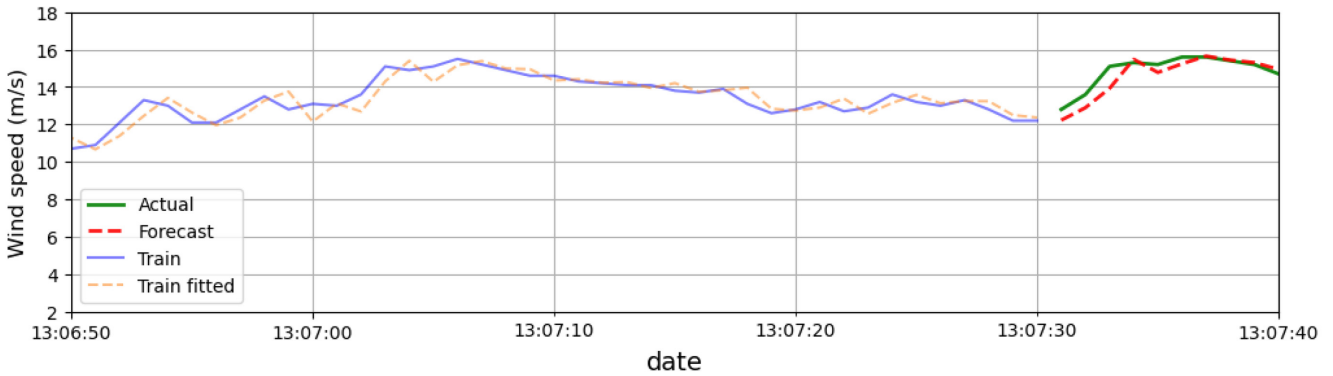
7. Conclusion

In this research, we compare several machine learning models to make short-term predictions on wind speed using wind

speed data obtained at wind turbine site. This prediction is to operate safely and prevent possible damage due to sudden gust.

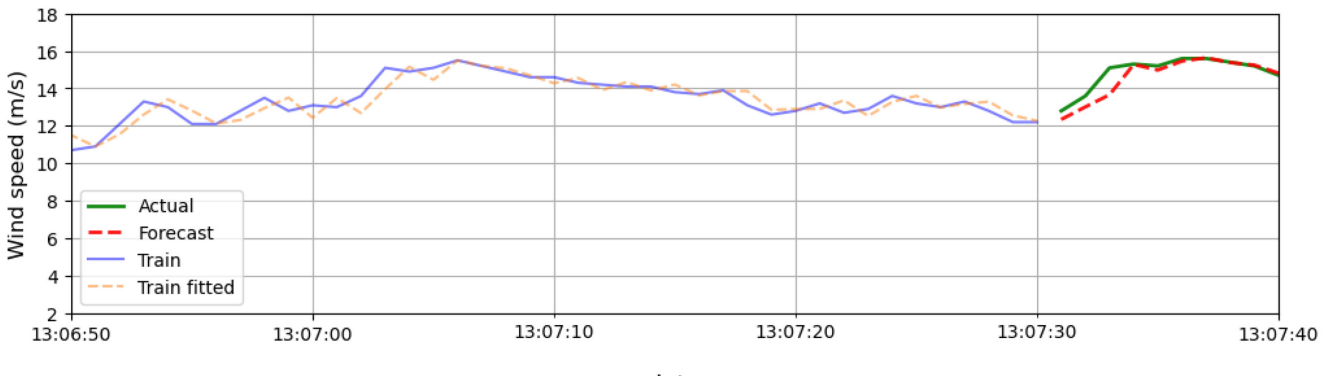
We implemented artificial neural networks in making a time series forecast. Three models that we used are recurrent neural

Model: RNN (train 2), RMSE=0.244, MAE=0.1836



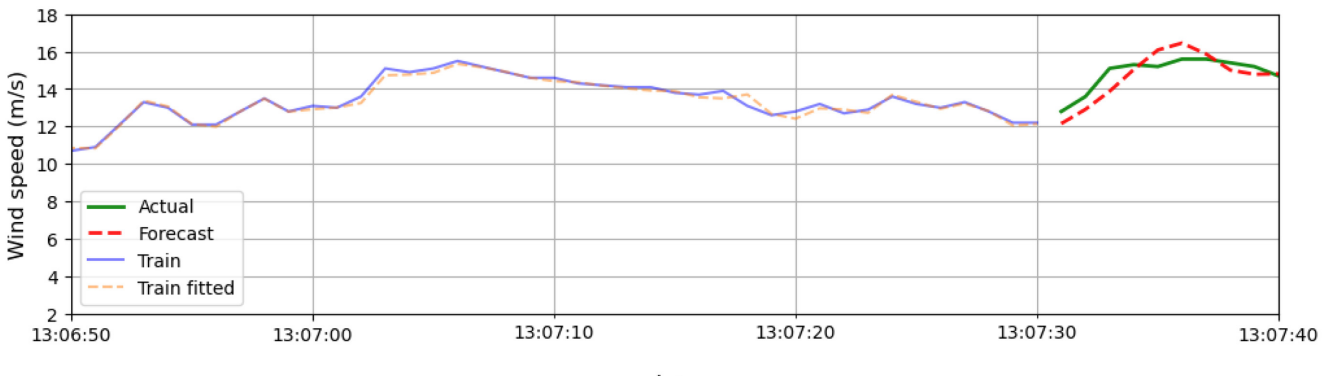
(a)

Model: LSTM (train 2), RMSE=0.2477, MAE=0.1468



(b)

Model: GRU (train 2), RMSE=0.3153, MAE=0.2737



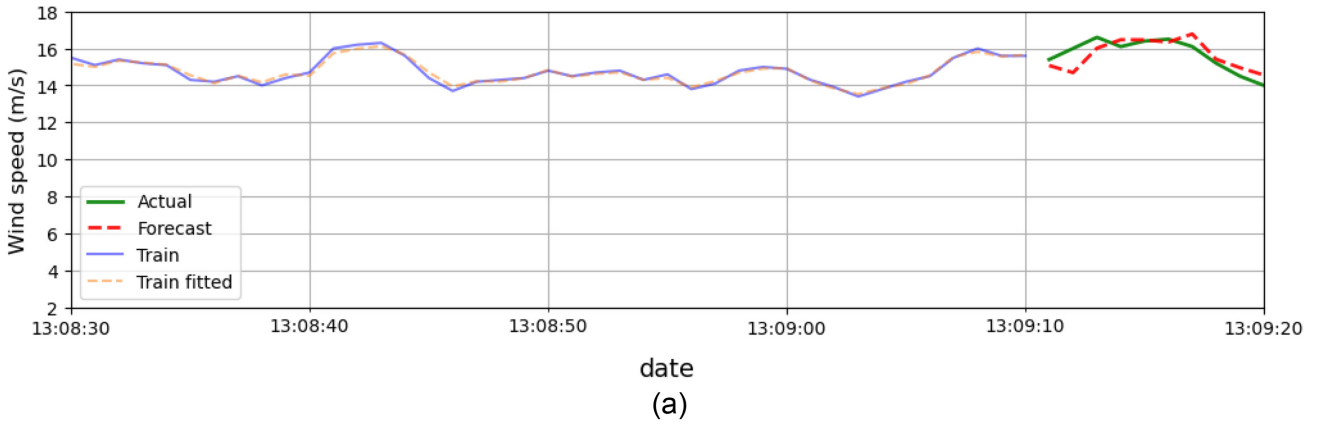
(c)

Fig. 10. (a) Case 6. Test Evaluation (Forecast at 13:07:30), (b) Case 6. Test Evaluation (Forecast at 13:07:30), (c) Case 6. Test Evaluation (Forecast at 13:07:30).

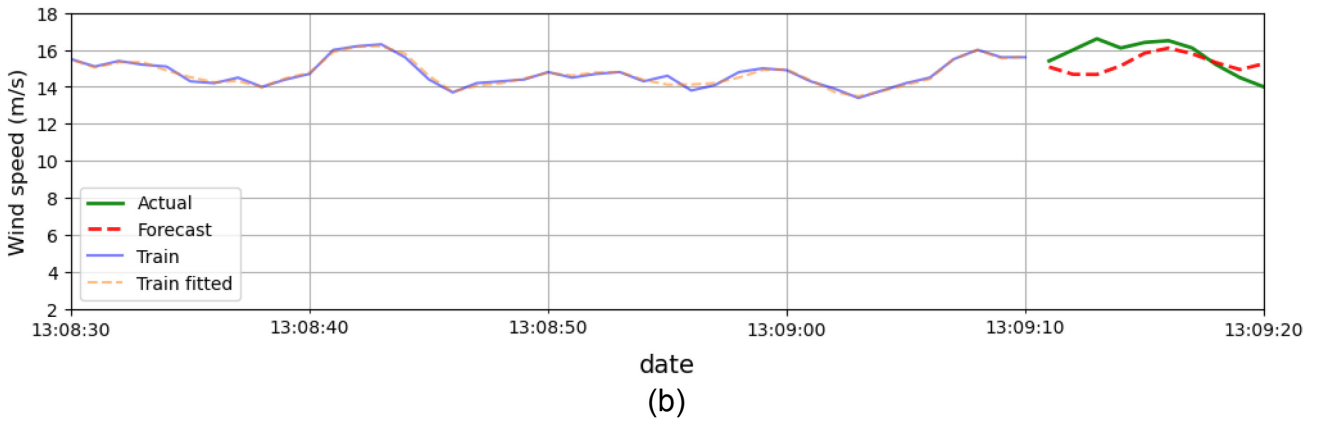
network (RNN), long sort-term memory (LSTM), and gated recurrent unit (GRU). All models are implemented using TensorFlow and Keras.

The wind data that is used for the analysis is resampled at 1 Hz. To train models 500 epochs are used with batch size 32 for most of the cases. Adams optimizer is used. And the loss is

Model: RNN (train 2), RMSE=0.6785, MAE=0.5527



Model: LSTM (train 2), RMSE=1.101, MAE=0.8921



Model: GRU (train 2), RMSE=0.978, MAE=0.7663

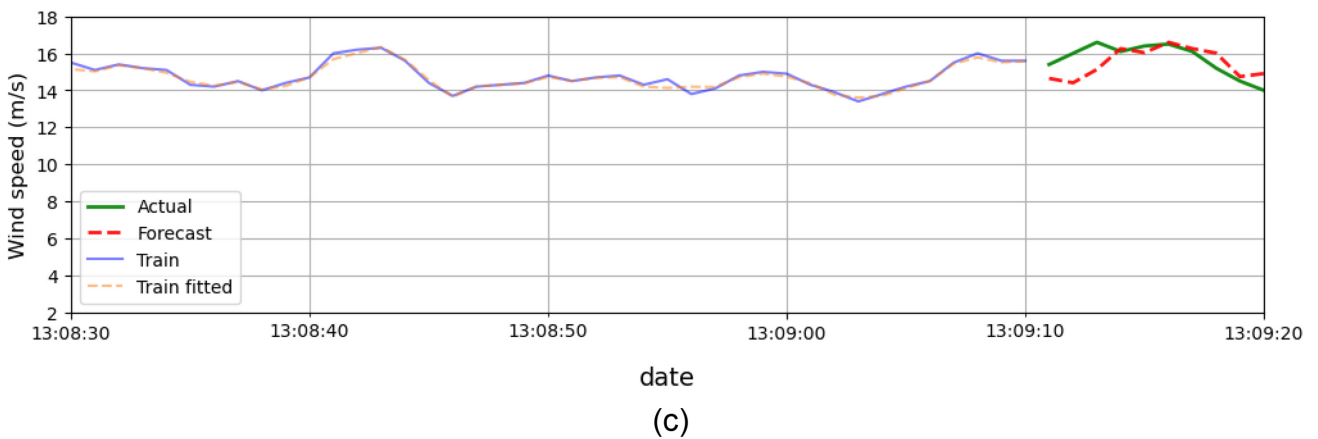
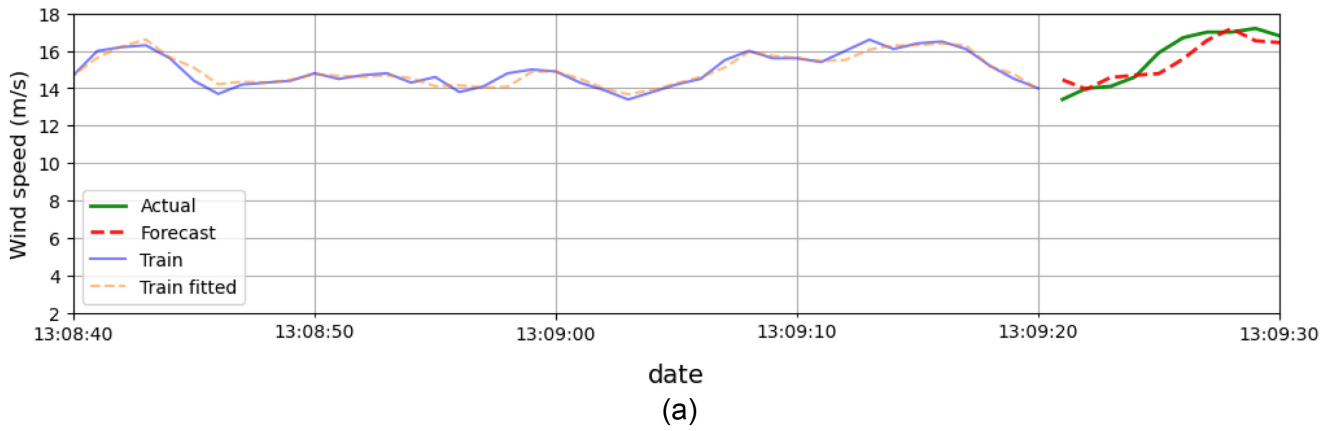


Fig. 11. (a) Case 7. Test Evaluation (Forecast at 13:09:10), (b) Case 7. Test Evaluation (Forecast at 13:09:10), (c) Case 7. Test Evaluation (Forecast at 13:09:10).

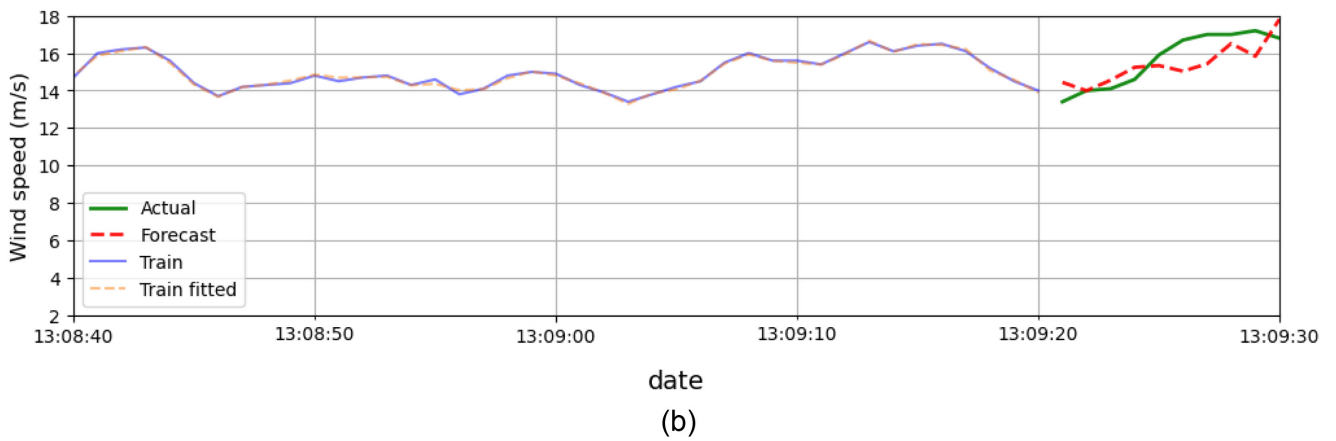
based on root mean square error and mean absolute error. 20% dropout is applied. We used 128 hidden layers for computation. Wind data are segmented into 120 second intervals. The first

110 seconds are used for training data while the final 10 seconds are used for test data. Numerical computations were made on two regions that

Model: RNN (train 2), RMSE=0.8278, MAE=0.6742



Model: LSTM (train 2), RMSE=1.2322, MAE=1.0648



Model: GRU (train 2), RMSE=1.0224, MAE=0.7888

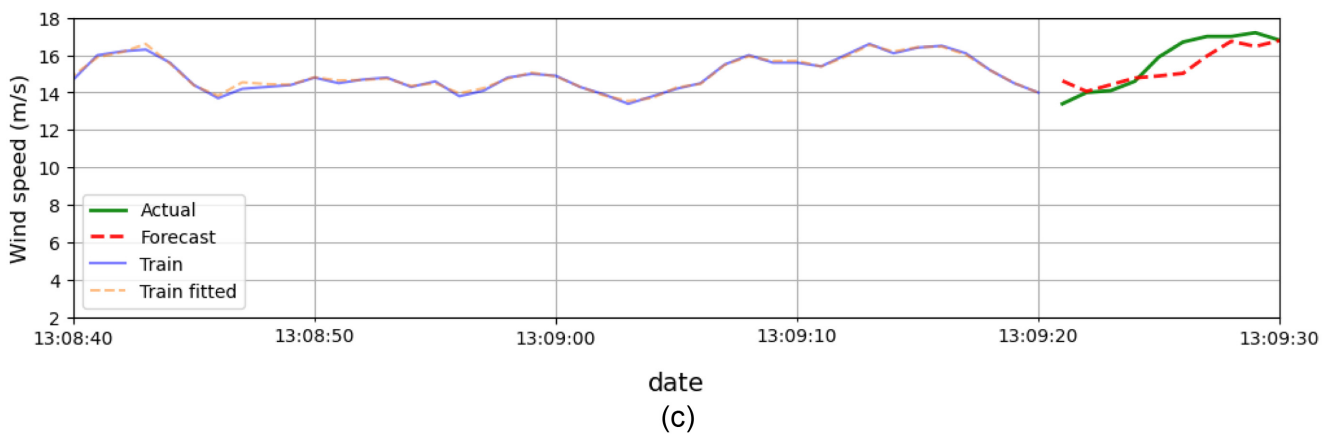


Fig. 12. (a) Case 8. Test Evaluation (Forecast at 13:09:20), (b) Case 8. Test Evaluation (Forecast at 13:09:20), (c) Case 8. Test Evaluation (Forecast at 13:09:20).

include rapidly changing wind speed span and the maximum wind speed span. Eight cases of prediction results from two regions were presented. All three artificial neural network models were

fitted with the training data well. Forecasting results generally follow the wind speed and the changing pattern of the wind speed. Among the three models, the RNN model predicts wind

speed relatively better than other models when the wind speed is changing rapidly in a short duration. This is likely because the RNN structure is simpler than other models, allowing it to respond more immediately to wind speed predictions that change in a very short period of time. The RNN model is the most efficient model in computing time since the number of parameters to fit is less than other models.

References

- [1] De Giorgi, M.G., Ficarella, A. and Tarantino, M., 2011, Error analysis of short-term wind power prediction models. *Applied Energy*, 88, 129-140.
- [2] GWEC, 2019, Global Wind Report 2018; Global Wind Energy Council (GWEC): Brussels, Belgium.
- [3] Hanifi, S., Liu, X., Lin, Z. and Lotfian, S., 2020, A Critical Review of Wind Power Forecasting Methods—Past, Present and Future. *Energies*, 13(15) 3764.
- [4] Hong, T., Zhang, W. and Zhang, S., 2018, Wind Speed Prediction Using Machine Learning Techniques. *Energies*, 11(11), 3100.
- [5] Kim, T., 2020, A Study on the Prediction Technique for Wind and Wave Using Deep Learning, *J. Korean Soc. Mar. Environ. Energy*, 23(3), 142-147.
- [6] Lange, M. and Focken, U., 2006, *Physical Approach to Short-Term Wind Power Prediction*. Springer, Berlin, Germany.
- [7] Lazzerii, F., 2020, *Machine Learning for Time Series Forecasting with Python*, Wiley.
- [8] MATLAB, Long Short-Term Memory Neural Networks, <https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html>, 2025, (accessed 2025.07.04)
- [9] Simplilearn, Power of Recurrent Neural Networks (RNN): Revolutionizing AI, <https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn>, 2025, (accessed 2025.07.04)
- [10] Techopedia, Gated Recurrent Unit, <https://www.techopedia.com/definition/33283/gated-recurrent-unit-gru>, 2025, (accessed 2025.07.04)
- [11] Yang, X.Y., Zhang, Y.F., Yang, Y.W. and Lv, W., 2019, Deterministic and Probabilistic Wind Power Forecasting Based on Bi-Level Convolutional Neural Network and Particle Swarm Optimization. *Appl. Sci.* 9, 1794.
- [12] Zhang, S.H., Liu, Y.W., Wang, J.Z. and Wang, C., 2019, Research on Combined Model Based on multi-objective optimization and application in wind speed forecast. *Appl. Sci.* 9, 423.
- [13] Zhang, S., Wang, J. and Wang, C., 2019, Research on combined models for wind speed forecasting based on optimization techniques. *Applied Sciences*, 9, 423.
- [14] Zhao, X., Wang, S. and Li, T., 2011, Review of evaluation criteria and main methods of wind power forecasting. *Energy Procedia*, 12, 761-769.

Received 5 August 2025

1st Revised 13 August 2025, 2nd Revised 21 February 2026

Accepted 22 February 2026