Leveraging multi-head attention transformer deep neural network architecture for improved wind speed forecasting

Nguyen Thi Hoai Thu*, Nguyen Trung Tuan Anh, Pham Phong Ky

PGRE. Lab., School of Electrical and Electronic Engineering, Hanoi University of Science and Technology
Corresponding author E-mail: thu.nguyenthihoai@hust.edu.vn
DOI: https://doi.org/10.64032/mca.v29i2.283

Abstract

Wind energy has great potential for electricity generation, but its variability makes accurate wind speed forecasting essential for efficient integration. This study explores the application of a transformer-based deep learning model for wind speed forecasting. The model features an encoder-decoder architecture with multi-head attention, feed-forward layers, and normalization functions. By leveraging a self-attention mechanism, the transformer model effectively captures temporal dependencies in time series data through weighted relationships among input sequences, leading to improved forecasting accuracy. To evaluate its effectiveness, we collected and pre-processed wind speed data from the Hong Phong 1 wind power plant, cleaned the data by removing outliers and addressed missing values. The processed data was then embedded and added positional encoding to prepare for model input. The model was trained, and its performance was benchmarked against other models, including Long Short-Term Memory, Convolutional Neural Networks, and Artificial Neural Networks. The obtained RMSE is quite low, with 0,26 m/s for single-step forecast, 0,73 m/s for 4-step forecast and 1,70 m/s for 16-step forecast. These results demonstrated that the transformer model achieved superior predictive performance, suggesting it as a powerful alternative to traditional forecasting methods, with significant potential for enhancing the accuracy of wind speed predictions.

Keywords*: Wind speed forecast; Deep learning; Transformer; Self-attention; Multi-step forecasting.*

Abbreviations

CNN	Convolutional Neural Network
ANN	Artificial Neural Network
LSTM	Long Short-Term Memory
FF	Feed-forward layer
MSE	Mean Square Error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
NRMSE	Normalized Root Mean Square Error

1. Introduction

Energy demand continues to rise in the modern era, yet traditional sources like coal, oil, and gas are not only depleting but also major contributors to environmental pollution. Therefore, developing sustainable energy alternatives is essential. Wind energy, a clean and abundant resource, addresses pollution concerns while offering notable economic advantages. The Global Wind Power Report 2022 notes that by the end of 2021, global wind power capacity reached 837 GW [1]. By 2040, wind power is expected to generate approximately 8,300 TWh, surpassing solar PV at 7,200 TWh and hydropower at 6,950 TWh [2]. However, the variability and intermittency inherent to wind power pose challenges for balancing supply and demand. Accurate wind speed forecasting is crucial for reliable wind power output prediction, reducing uncertainty, enhancing system stability, and ensuring power quality. Thus, precise wind speed forecasting has emerged as a critical area of research.

The wind speed forecast discussed in this paper focuses on predicting future wind speed values at specific points. It mainly includes physical forecasting models, statistical models and artificial intelligence (AI) models. Physical methods, often based on lower atmospheric dynamics or numerical weather prediction (NWP), use weather-related data like temperature, pressure, surface roughness, and local obstacles [3]. These models are particularly effective for medium- and long-term forecast [4], but may not perform well when dealing with real data that involves complex relationships. Statistical models, which include time series models like autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), predict wind speed based on historical data. AR model uses linear functions to predict values from historical data, while the MA model represents the current value of the time series as a function of past noise terms. Meanwhile, the ARMA model combines two main components: AR and MA [5], ARIMA is made up of AR and MA along with an integrated component (I). However, these models, with their focus on linear analysis and stable data inputs, are often less suitable for highly nonlinear time series data [6]. Additionally, statistical models also face limitations such as reduced accuracy in long-term forecasting, poor performance with seasonally dependent time series, and reliability concerns. Recently, AI methods have gained prominence due to their ability to handle complex systems and improve prediction accuracy [7]. Machine learning, deep learning, and fuzzy logic offer significant flexibility, with techniques like neural networks being effective for modeling nonlinear data and identifying complex relationships [8]. Nevertheless, these models demand considerable computational power, pose training challenges, and are susceptible to overfitting. Deep Learning (DL), a subset of machine learning, exhibits advanced learning capabilities, with models such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Convolutional

Received: 24 February 2025; Revised: 18 April 2025; Accepted: 18 April 2025.

Neural Networks (CNN) demonstrating notable success. However, these models often face challenges in capturing long-term dependencies. While models like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) can address these challenges, they struggle with efficient feature extraction and often have slower training times [9]. Artificial Neural Network (ANN) is capable of handling the modeling of complex systems [10]. However, the disadvantages of ANN network are that it requires large amounts of data and is at risk of overfitting.

The transformer is an advanced deep learning model that has gained significant prominence in forecast area [11]. Built on the attention mechanism, it serves as a highly effective sequence transcription tool, adept at overcoming many limitations found in traditional approaches [12]. Unlike sequential models, the transformer processes input sequences in parallel, enhancing both computational efficiency and accuracy [13]. The encoder captures and encodes these intricate patterns, while the decoder produces predictions based on the information learned by the encoder. The transformer has demonstrated its potential across various fields, including natural language processing, computer vision, and energy forecasting, marking it as a promising tool for both industrial applications and academic research [14].

In this paper, we used a transformer model for wind speed prediction using data from the Hong Phong 1 power plant. The primary aim was to forecast future wind speed values and evaluate the transformer model's performance against traditional models like LSTM, CNN, and ANN. This model was selected for its strength in modeling long-term dependencies and complex temporal patterns, which are essential characteristics of wind speed time series. The model configuration, including the number of attention heads and layers, was carefully adjusted to fit the time series data and the specific requirements of wind speed forecasting. To enhance model generalization and mitigate overfitting, we applied dropout after each key sub-layer, especially when integrating positional encoding. This approach helped reduce positional dependence, improving the model's adaptability and robustness for forecasting tasks.

The remainder of the paper is organized as follows. Section 2 covers the methodology used in this study. The results and discussion are presented in Section 3. And finally, Section 4 provides the conclusion.

2. Methodology

2.1. Transformer structure

The transformer architecture, as illustrated in Figure 1 [15], consists of an encoder on the left and a decoder on the right. The encoder is composed of two primary sub-layers: a multi-head attention mechanism and a feed-forward neural network [16]. Similar to the encoder, the decoder also includes these two layers but incorporates an additional layer called masked multi-head attention. Between each of these layers, there is an "add & normalize" component.

2.1.1. *Input embedding*

First, the data undergoes embedding. In wind speed dataset, information about position of the data in the set is important because it indicates the level of wind speed during that specific time period. However, the transformer model lacks an inherent mechanism to represent the positional information of the input sequence. Therefore, positional information must be incorporated by adding positional encoding to the embedded input. Both the input embedding and positional encoding share the same dimension d_{model}. The positional encoding is represented as follows:

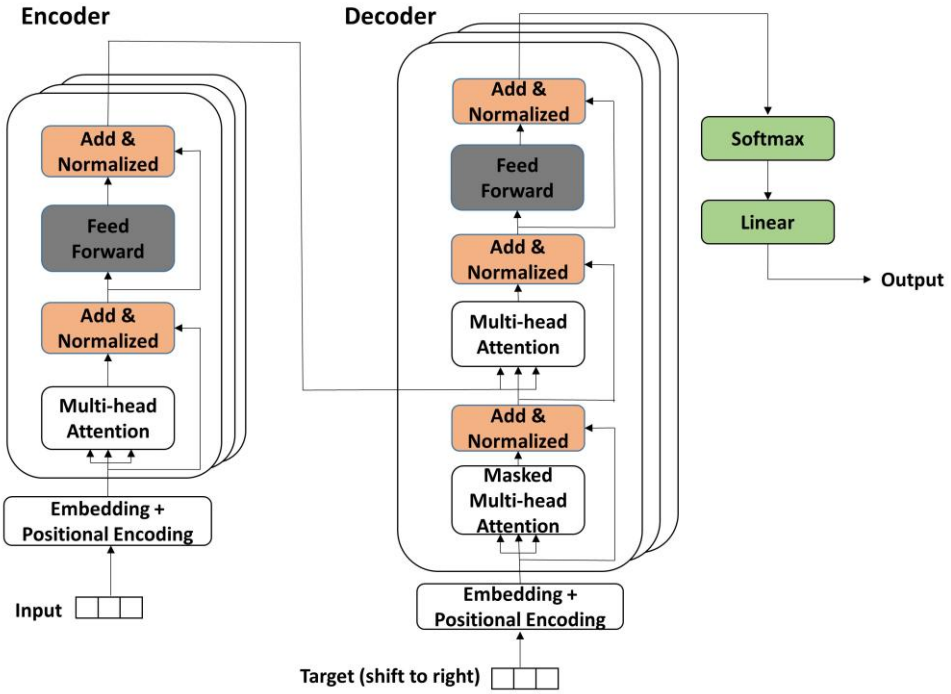

Figure 1: Architecture of transformer model

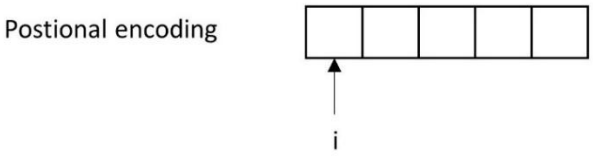

Figure 2: Positional encoding

$$PE_{(pos,i)} = \sin\left(\frac{pos}{10000^{\frac{i}{d_{model}}}}\right) \quad (1)$$

$$PE_{(pos,i)} = \cos\left(\frac{pos}{10000^{\frac{i-1}{d_{model}}}}\right) \quad (2)$$

where pos is the position of the data in the sequence data, i is the i-th dimension of positional encoding. Equation (1) is used when i is even and equation (2) if i is odd. The i value of positional encoding is determined as shown in Figure 2.

2.1.2. *Encoder*

Suppose the set of input vector is X={X_1, X_2, X_3,…X_n), the encoder maps X to head H={H_1, H_2, H_3,…H_n}. The encoder consists of two primary sub-layers: a multi-head attention mechanism and a feed-forward neural network. The multi-head attention layer is combined with dropout. Each sub-layers uses a residual structure and then the output data is layer-normalized [17]. They can be expressed as:

$$z = \gamma \times \frac{(x + Outsublay(x)) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (3)$$

where x is the input of a sublayer and Outsublay (x) is the output of the sublayer with input x, μ is the mean of (x+Outsublay(x)), σ^2 is the standard deviation of (x+Outsublay(x)), ϵ small constant to avoid division by zero, γ and β are the learnable parameters (scale and shift) of normalization layers, z is the normalized output.

2.1.3. *Multi-head attention layer*

How the multi-head attention mechanism works is illustrated in Figure 3. The input vectors are combined into a matrix of size (n, d_{model}) with n is number of input vectors, then vector X is multiplied with the weight matrices W_Q, W_K, W_V of size (d_{model}, d_{model}) to get matrix Q, K, V has size (n, d_{model}). Then the vectors will be divided into h representing h heads Q_i, K_i, V_i of size (n, d_{model}/h) by multiplying with the matrices W_q^i, W_k^i, W_v^i of size (d_{model}, d_{model}/h). Then perform the attention calculation for each head [18]:

$$Attention(h_i) = softmax\left(\frac{\mathbf{Q}_i \cdot \mathbf{K}_i^T}{\sqrt{d_k}}\right) \quad (4)$$

$$head_i = Attention(h_i).V_i \quad (5)$$

Where d_k=d_{model}/h.
Then group the outputs of the heads together via the concat function [19]:

$$Multihead = Concat\ (head_1, \dots, head_h) \quad (6)$$

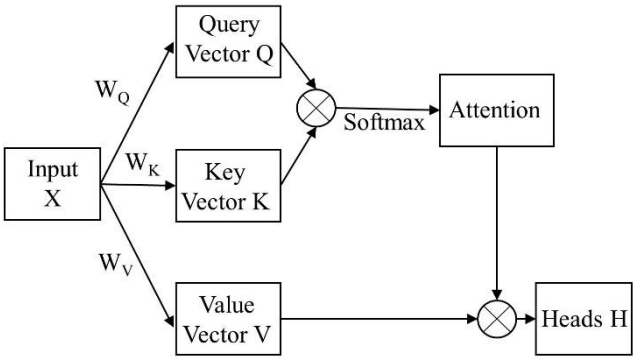

Figure 3: Multi-head attention operation

2.1.4. *Feed- forward layer (FF)*

The FF network uses two linear transformation matrices along with a Rectified Linear Unit (ReLU) activation function. Because of the existence of two linear transformation matrices, the dimension of the output of this layer is kept equal to d_{model} while the dimension can be adjusted for calculation within the layer. The formulation is as follow [20]:

$$FF(x) = max(0, zW_1 + b_1)W_2 + b_2 \quad (7)$$

where z is the output of add & normalization layers, W_1 is the linear transformation matrix with size (d_{model}, d_{ff}) and W_2 is the matrix with size (d_{ff}, d_{model}).

2.1.5. *Decoder*

The final input of the encoder input serves as the starting for the decoder input. The transformer's decoder consists of several decoder blocks, each block is made up of the same two sub-layers as the encoder blocks, with an extra sub-layer called the masked multi-head attention layer. Similarly to encoders, decoders also feature residual connections and normalization layers following each sub-layer [21]. Multi-attention and feed-forward layers operate in the same way as two layers of the same name in the encoder but in multi-head attention layer the matrix which multiple with W_K, W_V to get matrix K, V is the output of encoder. The special thing here is that the way mask-multihead attention layers operations is similar to how multi-head layer works but, but the input here is the output target sequence and:

$$Mask\ Scores = Attentions + Mask \quad (8)$$

$$Head = \text{softmax}(Mask\ Scores).V \quad (9)$$

Where Mask is a main diagonal matrix with the upper half having a value of $-\infty$ and the lower half having a value of 0.

2.2. Experimental details

2.2.1. *Dataset collection*

In this experiment, the historical wind speed dataset from Hong Phong 1 wind power plant was collected during the period from February 1st, 2022, to December 31st, 2022, which includes actual measurements of wind speed (m/s) with

sampling time of 15 minute. Hong Phong 1 wind power plant is located in Binh Thuan province, Vietnam. Binh Thuan province is located on the coast of the South-Central Coast region, has a sub-equatorial tropical monsoon climate, not much influenced by the Northeast monsoon, so it is a province with lots of sunshine, lots of wind, many storms and has quite high humidity. That area has two distinct seasons: the rainy season lasts from May to October, and the dry season lasts from November to April of the following year. Binh Thuan possesses diverse wind regimes and is one of the locations with the greatest potential for wind power development in Vietnam. The Northeast monsoon is common during the dry season and the Southwest monsoon blows heavily during the rainy months. Additionally, 80% of the dataset was used as the training set, the next 10% acted as the validation set, and the rest was the test set.

2.2.2. *Data preprocessing*

The experimental data may have outliers or missing values and needs to be preprocessed. The outliers were found and replaced, missing values eliminated if not needed or filled in by interpolation technique. After processing outliers and missing values, having too many large data values in the data set can impact the results of data analysis. To rescale different features, min-max normalization was used [22]:

$$x^* = \frac{(high-low)(x-min)}{max-min} \quad (10)$$

Where max, min are, respectively, the largest and smallest value of data set. The data scaled down to a range [0,1] so high euqualed to 1 and low equaled to 0. Normalization ensures that input variables have similar ranges and distributions, which may contribute to the models learning.

2.2.3. *Model hyper-parameters*

In order to evaluate the forecasting performance, four models of transformer, LSTM, CNN and ANN were built and utilized to predict the wind speed with different horizons of single-step, 4-step and 16-step. The transformer model adds positional information to the embedded input using positional encoding. The self-attention layers enable the model to learn and analyze relationships between data in a set, without considering the distance between them. To learn many types of relationships between data, the model uses multi-head attention layers with many self-attentions. Then the outputs of multi-head attention layers pass through a series of FF layers. By using these layers, the model can learn complex non-linear relationships between the input and output and FF layers also help enhance the model's capacity [23]. The decoder of the transformer model for time series forecasting uses a mask to make the model learn to predict the future data without information about the following data. We used the prediction horizon of single-step, 4-step and 16-step. The hyperparameters set for the transformer model as well as the compared models are shown in Table 1. These hyperparameters were tuned via Optuna framework to obtain the most optimal hyperparameter sets.

2.2.4. *Evaluation*

In this paper, three metrics were used for evaluation: Mean square error (MSE), mean absolute error (MAE) and root mean square error (RMSE). The MSE measures the average squared difference between the predicted and actual values, the lower the MSE, the higher the model's accuracy. The MAE calculates the absolute of the average differences between the predicted values and the actual values. And it indicates the distance between the forecast value and the actual value. The RMSE demonstrates the magnitude of the average error between actual and forecasted values and offers

Table 1: Important parameters of the all models

Model	Hyper-parameter	1-step	4-step	16-step
Transformer	Embedding dimension (d_{model})	32	40	64
	Number of heads (h)	2	2	4
	Number of layers	1	1	1
	Dropout	0,1	0,1	0,15
	Learning rate (lr)	0,0025	0,0014	0,002
LSTM	Hidden layers	3	2	3
	Neurons in layer	80	150	233
	Activation	ReLU	ReLU	ReLU
	Optimizer	Adam	Adam	Adam
	Dropout	0,1	0,16	0,11
	Learning rate (lr)	0,003	0,0036	0,0012
ANN	Hidden layers	3	3	3
	Neurons (each layer)	(48,128,64,1)	(48,200,100,4)	(48,334,208,16)
	Activation	ReLU	ReLU	ReLU
	Dropout	0,1	0,1	0,12
	Learning rate (lr)	0,0015	0,001	0,0051
CNN		Activation shape	Activation shape	Activation shape
	Input	(64,48,1)	(68,48,1)	(64,48,1)
	Conv1	(64,43,50)	(64,46,83)	(64,46,55)
	Conv2	(64,42,100)	(64,44,206)	(64,44,214)
	MaxPooling	(64,21,100)	(64,22,206)	(64,22,100)
	Flatten	(64,2100)	(64,4532)	(64,4708)
	Output	(64,2100,1)	(64,4532,1)	(64,4708,1)
	Dropout	0,1	0,1	0,018
	Learning rate (lr)	0,0015	0,0004	0,0022

the same measurement scale as the actual results. They can be expressed as:

$$MSE = \frac{1}{l}\sum_{i=1}^{l}\left(p - \hat{p}\right)^2 \quad (11)$$

$$MAE = \frac{1}{l}\sum_{i=1}^{l}\left|p - \hat{p}\right| \quad (12)$$

$$RMSE = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\left(p - \hat{p}\right)^2} \quad (13)$$

where p denotes the actual value, $\hat{p}$ denotes the predicted value, l denotes the length of predicted series.
Moreover, NRMSE was also used to normalize RMSE, allowing comparison of error levels between different models on data sets with different scales. It can be expressed as:

$$NRMSE = \frac{RMSE}{\max(y) - \min(y)} \quad (14)$$

where max (y) is the max value of actual set, min (y) is the min value of actual set.

3. Results and Discussion

The results provided a comparative evaluation of transformer, LSTM, CNN, and ANN models in the forecasting task, which aimed to predict a sequence of future values based on current data. Four performance metrics were reported: MSE, MAE, RMSE and NRMSE. All four models were trained on the same training set with different forecast steps, using the same computational environment with identical hyperparameters for their training. We fine-tuned the proposed model, and the models used for comparison to ensure that the models performed with the best-fit set of hyperparameters. Table 2 presents the detailed performance evaluation metrics for the transformer, LSTM, CNN, and ANN models.

3.1. Single-step forecasting

For single-step forecasting, it is clear from Table 2 that the transformer model has better performance than the other models when the transformer shows quite low errors, with values such as MSE at 0,07, MAE at 0,18, RMSE at 0,26, and NRMSE at 0,02. Compared to ANN (the model with the largest MSE error in single-step forecasting), the value of MSE error of the transformer model is 0,07, which is only 16,28% of the equivalent value of the ANN model. The corresponding ratios compared to the LSTM and CNN models are 17,95% and 16,28%, respectively. Since the MSE indicator highlights the effect of large errors, the fact that the proposed model has a small MSE value indicates that the model has no or a negligible amount of error between the actual value and the predicted value. The MAE value of the transformer model is 0,18, which is also the lowest while the MAE value of the ANN model is still the highest. This means that on average, the model's wind speed forecast only differs by about 0,18 unit from the actual value, which shows that the forecast value fits reality quite precisely. This indicator of the transformer model is only about half of that of the other models. The same is seen in the RMSE and NRMSE indicators, where the values of the transformer model are still lower than all three models LSTM, CNN and ANN. These results demonstrate that the transformer model performs very well in single step forecasting and outperforms the compared models. Figure 4 shows the single-step wind speed forecasting results of four models.

3.2. Multi-step forecasting

A similar trend is also seen in multi-step forecasting. Multi-step forecasting is performed including 4-step forecasting and 16-step forecasting. Although the difference between the values of indicators of the models decrease, in general the indicators of the transformer model still have the lowest value. The MSE error of the transformer model in 4-step forecasting

Table 2: Performance evaluation indicators of LSTM, CNN, ANN and transformer model

Step	Model	MSE (m/s)²	MAE (m/s)	RMSE (m/s)	NRMSE
Single-step	LSTM	0,39	0,41	0,63	0,04
	CNN	0,43	0,44	0,66	0,04
	ANN	0,43	0,42	0,65	0,04
	Transformer	**0,07**	**0,18**	**0,26**	**0,02**
4-step	LSTM	1,08	0,72	1,04	0,07
	CNN	1,09	0,75	1,04	0,07
	ANN	1,06	0,72	1,03	0,07
	Transformer	**0,23**	**0,33**	**0,49**	**0,03**
16-step	LSTM	2,76	1,25	1,66	0,11
	CNN	2,75	1,25	1,66	0,11
	ANN	2,88	1,27	1,69	0,12
	Transformer	**0,59**	**0,49**	**0,77**	**0,05**

is only about 21% of other models while the MAE, RMSE and NRMSE errors only fluctuate within 40-50% of the corresponding values for the other models. In the 16-step forecasting problem, the transformer model still achieves the best results despite the significant increase in steps with MAE of 0,49 m/s and RMSE of 0,77 m/s, significantly lower than LSTM, CNN and ANN (all above 1,2 m/s). These indicate that transformer model is better even in multi-step forecasting. Figure 5 and Figure 6 present the results of 4-step and 16-

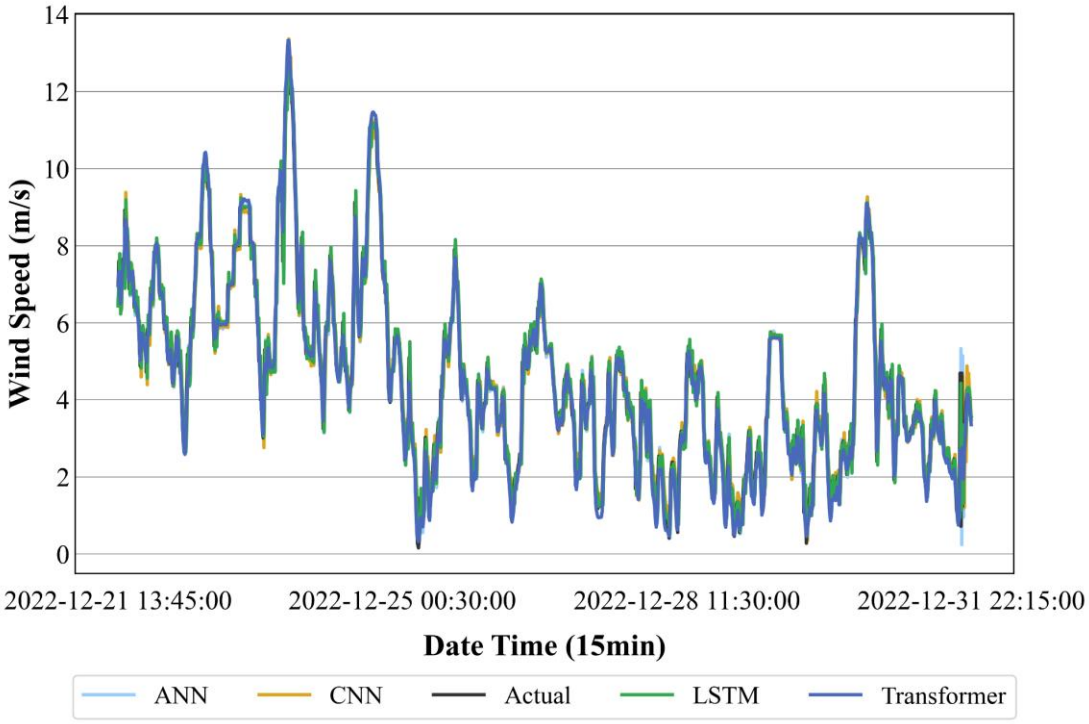

Figure 4: Single-step wind speed forecast results at Hong Phong 1

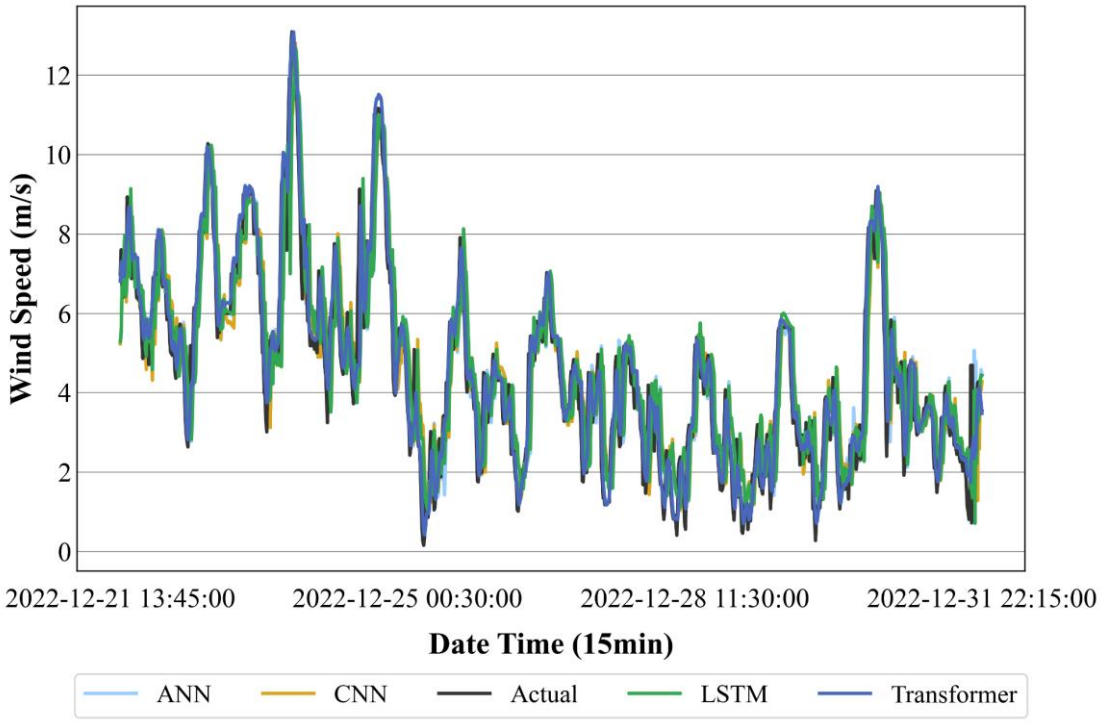

Figure 5: 4-step wind speed forecast results at Hong Phong 1

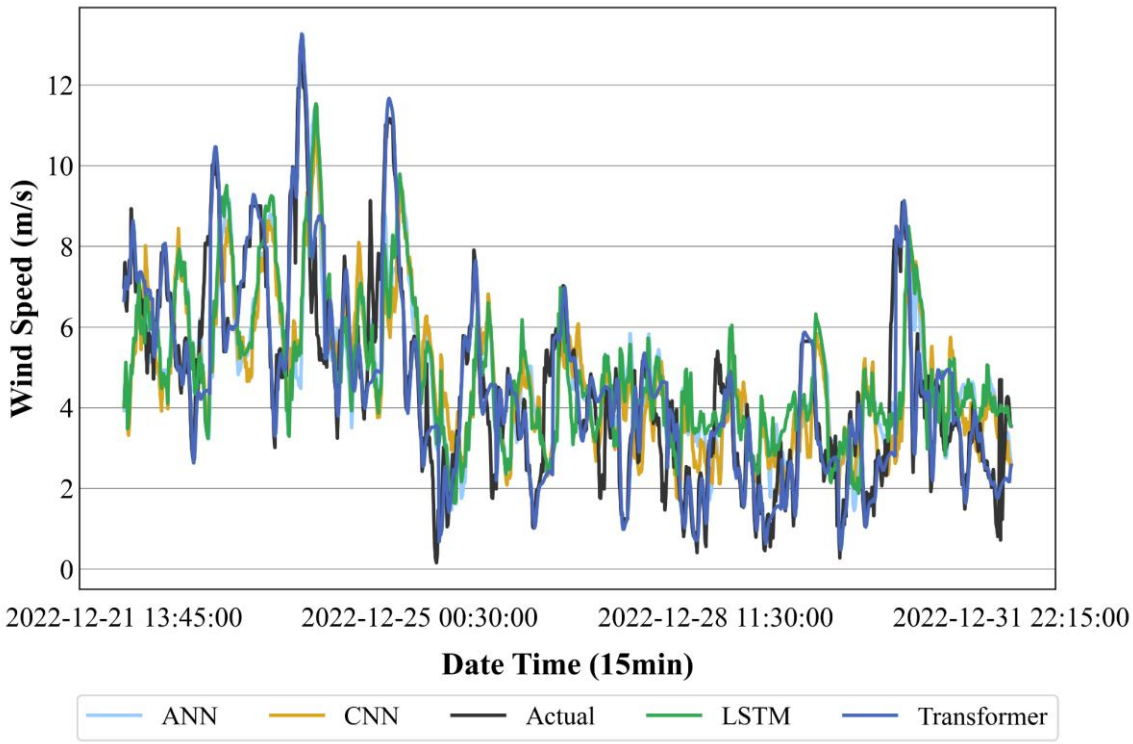

Figure 6: 16-step wind speed forecast results at Hong Phong 1

step wind speed forecasting results of the four models, respectively.

3.3. Discussion

In every instance, the transformer model outperforms the other models across various prediction intervals. As expected, the CNN and ANN models perform worse than both the LSTM and transformer models. Even when unusual data are present in the dataset, the performance of the transformer model only drops slightly compared to other models. This demonstrates that the proposed model exhibits better adaptability to weather changes and excellent generalization capability, making it applicable to diverse datasets.

4. Conclusion

In this paper, we examined the structure and operational principles of the transformer model and compared its performance with LSTM, CNN, and ANN models. Utilizing historical wind speed data from the Hong Phong 1 wind power plant, we forecasted future wind speeds with both the proposed transformer model and the comparative models. The results demonstrated that the transformer model consistently outperformed the other models for both single-step and multi-step predictions. Specifically, the transformer model achieved the lowest values for MSE, MAE, RMSE, and NRMSE among all models with NRMSE only around 0,02 for 1 step, 0,03 for 4 steps and 0,05 for 4-hour ahead forecast. Although forecasting errors increased with the prediction horizon, the transformer model remained robust and reliable. Overall, the transformer-based approach demonstrated superior accuracy and effectively addressed the limitations of traditional forecasting methods.

Acknowledgments

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2024-PC-046

References

[1] "Global Wind Report 2022 - Global Wind Energy Council." Accessed: Jul. 20, 2024. [Online]. Available: https://gwec.net/global-wind-report-2022/

[2] N. N. V. Nhat, D. N. Huu, and T. N. T. Hoai, "Evaluating the EEMD-LSTM model for short-term forecasting of industrial power load: A case study in Vietnam," *Int. J. Renew. Energy Dev.*, vol. 12, no. 5, pp. 881–890, Sep. 2023, doi: 10.14710/ijred.2023.55078.

[3] W.-Y. Chang, "A Literature Review of Wind Forecasting Methods," *J. Power Energy Eng.*, vol. 02, no. 04, pp. 161–168, 2014, doi: 10.4236/jpee.2014.24023.

[4] T. H. T. Nguyen and Q. B. Phan, "Hourly day ahead wind speed forecasting based on a hybrid model of EEMD, CNN-Bi-LSTM embedded with GA optimization," *Energy Rep.*, vol. 8, pp. 53–60, Nov. 2022, doi: 10.1016/j.egyr.2022.05.110.

[5] D. Kaur, T. Tjing Lie, N. K. C. Nair, B. Vallès, and 1 Department of Electrical & Electronic Engineering, Auckland University of Technology, Auckland, New Zealand;, "Wind Speed Forecasting Using Hybrid Wavelet Transform—ARMA Techniques," *AIMS Energy*, vol. 3, no. 1, pp. 13–24, 2015, doi: 10.3934/energy.2015.1.13.

[6] N. Nhat, D. Nguyen Huu, and T. Nguyen, "Short-term multi-step forecasting of rooftop solar power generation using a combined data decomposition and deep learning model of EEMD-GRU," *J. Renew. Sustain. Energy*, vol. 16, Jan. 2024, doi: 10.1063/5.0176951.

[7] D. Bouabdallaoui, T. Haidi, F. Elmariami, M. Derri, and E. M. Mellouli, "Application of four machine-learning methods to predict short-horizon wind energy," *Glob. Energy Interconnect.*, vol. 6, no. 6, pp. 726–737, Dec. 2023, doi: 10.1016/j.gloei.2023.11.006.

[8] N. T. H. Thu, P. N. Van, N. V. N. Nam, and P. H. Minh, "Forecasting Wind Speed Using A Hybrid Model Of Convolutional Neural Network And Long-Short Term Memory With Boruta Algorithm-Based Feature Selection," vol. 26, no. 8.

[9] F. Tian, X. Fan, R. Wang, H. Qin, and Y. Fan, "A Power Forecasting Method for Ultra-Short-Term Photovoltaic Power Generation Using Transformer Model," *Math. Probl. Eng.*, vol. 2022, pp. 1–15, Oct. 2022, doi: 10.1155/2022/9421400.

[10] M. M. Ibrahim, A. A. Elfeky, and A. El Berry, "Forecasting energy production of a PV system connected by using NARX neural network model," *AIMS Energy*, vol. 12, no. 5, pp. 968–983, 2024, doi: 10.3934/energy.2024045.

[11] W. Jiang *et al.*, "Applicability analysis of transformer to wind speed forecasting by a novel deep learning framework with multiple atmospheric variables," *Appl. Energy*, vol. 353, p. 122155, Jan. 2024, doi: 10.1016/j.apenergy.2023.122155.

[12] P. C. Huy, N. Q. Minh, N. D. Tien, and T. T. Q. Anh, "Short-Term Electricity Load Forecasting Based on Temporal Fusion Transformer Model," *IEEE Access*, vol. 10, pp. 106296–106304, 2022, doi: 10.1109/ACCESS.2022.3211941.

[13] J. Lee, I. Bahk, H. Kim, S. Jeong, S. Lee, and D. Min, "An Autonomous Parallelization of Transformer Model Inference on Heterogeneous Edge Devices," in *Proceedings of the 38th ACM International Conference on Supercomputing*, in ICS '24. New York, NY, USA: Association for Computing Machinery, Tháng Sáu 2024, pp. 50–61. doi: 10.1145/3650200.3656628.

[14] P. Zhao *et al.*, "Enhancing multivariate, multi-step residential load forecasting with spatiotemporal graph attention-enabled transformer," *Int. J. Electr. Power Energy Syst.*, vol. 160, p. 110074, Sep. 2024, doi: 10.1016/j.ijepes.2024.110074.

[15] W. Li *et al.*, "An interpretable hybrid deep learning model for flood forecasting based on Transformer and LSTM," *J. Hydrol. Reg. Stud.*, vol. 54, p. 101873, Aug. 2024, doi: 10.1016/j.ejrh.2024.101873.

[16] E. G. S. Nascimento, T. A. C. De Melo, and D. M. Moreira, "A transformer-based deep neural network with wavelet transform for forecasting wind speed and wind energy," *Energy*, vol. 278, p. 127678, Sep. 2023, doi: 10.1016/j.energy.2023.127678.

[17] S. Reza, M. C. Ferreira, J. J. M. Machado, and J. M. R. S. Tavares, "A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks," *Expert Syst. Appl.*, vol. 202, p. 117275, Sep. 2022, doi: 10.1016/j.eswa.2022.117275.

[18] S. Xu, R. Zhang, H. Ma, C. Ekanayake, and Y. Cui, "On vision transformer for ultra-short-term forecasting of photovoltaic generation using sky images," *Sol. Energy*, vol. 267, p. 112203, Jan. 2024, doi: 10.1016/j.solener.2023.112203.

[19] S. F. Stefenon, L. O. Seman, L. S. A. Da Silva, V. C. Mariani, and L. D. S. Coelho, "Hypertuned temporal fusion transformer for multi-horizon time series forecasting of dam level in hydroelectric power plants," *Int. J. Electr. Power Energy Syst.*, vol. 157, p. 109876, Jun. 2024, doi: 10.1016/j.ijepes.2024.109876.

[20] E. Lezmi and J. Xu, "Time Series Forecasting with Transformer Models and Application to Asset Management," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4375798.

[21] H. S. Oliveira and H. P. Oliveira, "Transformers for Energy Forecast," *Sensors*, vol. 23, no. 15, p. 6840, Aug. 2023, doi: 10.3390/s23156840.

[22] S. Bhanja and A. Das, "Deep Neural Network for Multivariate Time-Series Forecasting," in *Proceedings of International Conference on Frontiers in Computing and Systems*, vol. 1255, D. Bhattacharjee, D. K. Kole, N. Dey, S. Basu, and D. Plewczynski, Eds., in Advances in Intelligent Systems Springer and Computing, vol. 1255. , Singapore: Singapore, 2021, pp. 267–277. doi: 10.1007/978-981-15-7834-2_25.

[23] E. Alerskans, J. Nyborg, M. Birk, and E. Kaas, "A transformer neural network for predicting near-surface temperature," *Meteorol. Appl.*, vol. 29, no. 5, p. e2098, 2022, doi: 10.1002/met.2098.