FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 250-256| e-ISSN: 2347-470X

Application of LSTM and GRU Neural Networks in Forecasting the Power Output of Wind Power Plant

Dan Bui Thi Tuyet[1*], Chau Le Thi Minh[2], Sa Nguyen Thi Mi[3], Duc Nguyen Tu[4], and Hieu Phu Thi Ngoc[5]

*[*1]Master of Engineering, Faculty of Electrical and Electronic Engineering, HCMC University of Technology and Education, Vietnam; danbtt@hcmute.edu.vn*

[2]Doctor of Philosophy, School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi; chau.lethiminh@hust.edu.vn

[3]Doctor of Philosophy, Faculty of Electrical and Electronic Engineering, HCMC University of Technology and Education, Vietnam; misa@hcmute.edu.vn

[4]Master of Engineering, Faculty of Electrical and Electronic Engineering, HCMC University of Technology and Education, Vietnam; ducnt@hcmute.edu.vn

[5]Master of Engineering, Faculty of Electrical and Electronic Engineering, HCMC University of Technology and Education, Vietnam; hieuptn@hcmute.edu.vn

***Correspondence:** *danbtt@hcmute.edu.vn*

ABSTRACT- This paper proposes the application of artificial intelligence to forecast the generation capacity of wind power plants by processing data through noise reduction and filtering. It subsequently employs Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) neural networks for training, testing, and evaluation. Processing the initial data will help minimize noise and reduce the data space. The study focuses on preprocessing methods and selecting the appropriate neural network between LSTM and GRU. The initial data processing will assess the similarity through the Spearman rank correlation coefficient. The data used in the paper is taken from local wind turbines. The processed data will be input into the neural network for evaluation based on Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Percent Error (PE), and training time. The entire network simulation and evaluation process is performed using MATLAB software. The simulation results show the feasibility and suitability of the GRU network model combined with noise filtering methods, bringing high accuracy and less training time compared to the LSTM network. Specifically, the analysis contrasting the GRU network with the optimized dataset and the LSTM with the unprocessed dataset is more effective than a difference in RMSE of 15.592 and MAPE of 611.047%. Moreover, the time difference between the GRU and LSTM networks with the same dataset has a much earlier time difference from 6 to 28 seconds.

Keywords: Forecasting Generator Power, Wind Power Plants, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Artificial Neural Network (ANN).

ARTICLE INFORMATION

Author(s): Dan Bui Thi Tuyet, Chau Le Thi Minh, Sa Nguyen Thi Mi, Duc Nguyen Tu, and Hieu Phu Thi Ngoc;

Received: 02/03/2025; **Accepted**: 30/04/2025; **Published**: 30/05/2025;

E- ISSN: 2347-470X;

Paper Id: IJEER 0203-01;

Citation: 10.37391/ijeer.130208

Webpage-link:

https://ijeer.forexjournal.co.in/archive/volume-13/ijeer-130208.html

Crossref member CROSSREF.ORG THE CITATION LINKING BACKBONE

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The increasing use of wind energy requires accurate forecasting models to ensure grid stability. However, traditional methods such as trend analysis [1] or econometric models [2], [3], [4] often fail to meet expectations due to the high variability of wind data, sudden weather changes, and complex environmental factors. This results in substantial forecasting inaccuracies, diminishing energy utilization efficiency, and compromising the reliability of the energy system [5].

Load forecasting is essential for grid operators as they develop strategies and manage oversight during their transition to more intelligent and more flexible energy systems. The integration of renewable energy sources, coupled with the increasing complexity of energy consumption patterns, necessitates accurate load forecasts to ensure reliability and efficiency in power supply. A recent study [6] has evaluated advancements in wind energy forecasting, focusing on how Artificial Intelligence (AI) and big data can enhance forecast accuracy.

The demand for advanced methods in wind power forecasting is becoming increasingly urgent [7]. Modern computational techniques offer improved adaptability to complex data variations, enabling more precise predictions. A model combining Temporal Convolutional Networks (TCN) and LSTM, integrated with the Savitzky-Golay filter, has been developed to reduce noise and improve accuracy under fluctuating weather conditions [8]. Similarly, an improved version of Wavenet leverages nonlinear features to enhance multi-step forecasting performance compared to ARIMA and linear regression [9]. Advanced models like LSTM and GRU are designed to handle complex time-series data, addressing the limitations of traditional methods in wind power forecasting, where climate factors constantly change and remain difficult to predict [10], [11].

LSTM and GRU excel in time-series processing by learning and retaining long-term relationships, a challenge for traditional models. LSTM utilizes gating mechanisms (forget, input, output) to filter and preserve crucial information, reducing noise and optimizing pattern recognition. GRU, a simplified version, lowers computational costs while maintaining high efficiency. In wind power forecasting, these models enhance accuracy and reliability, support the optimization of renewable energy systems, and ensure grid stability [12].

The performance evaluation process is crucial. Studies comparing LSTM, and GRU with traditional methods show that LSTM achieves higher accuracy based on RMSE, MAE, and MAPE [13]. The CNN-GRU hybrid model leverages spatial-temporal information, improving wind power forecasting [14]. Additionally, training time impacts feasibility in real-time forecasting. A rigorous data collection and preprocessing process ensures that LSTM and GRU operate optimally in renewable energy applications [15]. A forecasting model is evaluated based on standard error metrics such as RMSE, MAE, and MAPE, along with training time, ensuring accuracy and efficiency in energy forecasting [16].

This research focuses on data collection and preprocessing for wind turbine forecasting, using LSTM and GRU models to predict power generation and assess adaptability to weather variations. MATLAB is employed for simulation and validation, evaluating errors, training time, and processing speed. A model comparison is conducted to determine the optimal approach, enabling parameter adjustments for improved performance. Additionally, the study investigates and evaluates various data transformation techniques to identify which are most suitable for common neural network architectures in forecasting applications and yield the most accurate prediction results.

2. LITERATURE REVIEW

2.1. Limitations of Traditional Wind Power Forecasting Methods

Wind power forecasting plays a pivotal role in ensuring grid stability and facilitating the integration of renewable energy sources. However, traditional forecasting methods such as trend analysis [1] and econometric models [2–4] are often inadequate in capturing the inherent volatility and nonlinearity of wind data. These methods struggle to adapt to abrupt weather changes, leading to significant forecasting errors [5]. Despite their historical relevance, these techniques are increasingly being replaced by more dynamic and adaptable models that can better handle complex temporal and nonlinear characteristics.

2.2. Emergence of Machine Learning in Wind Forecasting

In response to the limitations of traditional models, machine learning techniques such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and ensemble learning have gained popularity in wind power prediction [6–7]. These approaches offer improved accuracy and adaptability, especially in capturing complex patterns.

Nevertheless, their effectiveness is often constrained by the need for extensive feature engineering and the inability to capture sequential dependencies present in time-series data, which limits their generalizability in dynamic environments.

2.3. Deep Learning Models: LSTM and GRU

Recent advancements have spotlighted deep learning models—specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)—as powerful tools for wind power forecasting. LSTM networks are designed to retain long-term dependencies through memory cells and gating mechanisms, making them highly effective in modeling temporal dynamics and filtering out noise [8–10]. GRUs, as a simplified variant of LSTM, offer comparable performance with fewer parameters, enabling faster computation and suitability for real-time applications [11–13].
These recurrent neural architectures mark a significant leap in handling the sequential nature of wind data, yet further investigation into their comparative strengths is still needed for practical deployment.

2.4. Hybrid and Filter-Integrated Deep Learning Approaches

To further enhance prediction robustness, hybrid models combining deep learning with signal processing techniques have been developed. For instance, integrating LSTM with filters like Savitzky-Golay or wavelet transforms has shown promise in mitigating noise and improving accuracy in rapidly changing conditions [8, 14].

Likewise, convolutional recurrent structures such as CNN-GRU have been explored for spatial-temporal modeling in multi-step forecasting tasks, yielding promising results in diverse scenarios [14–15].

Despite these improvements, such hybrid models often lack systematic evaluation frameworks and may vary widely in performance depending on data characteristics and preprocessing quality.

2.5. Research Gaps and Study Objectives

While numerous studies have explored the potential of deep learning and hybrid models in wind power forecasting, there remains a lack of standardized comparisons, especially concerning preprocessing pipelines and performance metrics. Moreover, few studies thoroughly investigate the role of data cleaning, noise filtering, and parameter optimization in enhancing model reliability.

This study addresses these gaps by:

- Applying and comparing LSTM and GRU models using a consistent data preprocessing pipeline;

- Utilizing MATLAB simulations to analyze model training time and accuracy *via* RMSE, MAE, and MAPE;
- Exploring the adaptability of deep learning models under varying weather conditions.

3. MATERIALS AND METHODS

3.1. Long-short term memory (LSTM)

LSTM is a specialized RNN designed to handle long-term dependencies in time series data while overcoming the vanishing gradient problem. Introduced by Hochreiter & Schmidhuber (1997), it has become widely used due to its effectiveness. Despite its complexity, LSTM retains the core sequential processing concept of RNNs.

A standard RNN network will have a relatively basic architecture consisting of one hidden layer using the tanh function, as shown below;

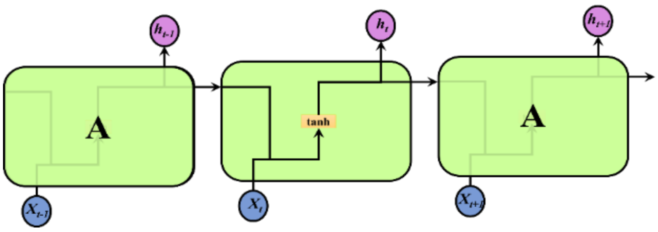

Figure 1. The repetition of the module architecture in an RNN network contains a hidden layer

LSTM similarly has a sequential structure, but its repeating architecture is more distinct. Instead of a single layer, it consists of 4 hidden layers (3 sigmoid and 1 tanh) that interact with each other in a specialized structure.

In *Figure 2,* the basic operation process of LSTM includes the following steps:

The first step in an LSTM is to decide which information we will allow to pass through the cell state. This is controlled by a sigmoid function in a layer known as the forget gate layer. First, it takes two input values, ht−1, and xt, and returns a value between 0 and 1 for each value of the cell state Ct−1. If the value is 1, it means "retain all information". If the value is 0, it means "discard all of it." The mathematical formula for this step is presented in *equation (1).*

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

The next step determines the type of information that will be stored in the cell state. This step consists of two parts. The first part is a hidden layer with a sigmoid function called the input gate layer, which decides the value to be updated. Next, the hidden tanh layer will create a vector of a new state value that can be added to the state. The following equation represents the computation process.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

Next, the results of these two layers are combined to create an update for the state.

In the example of a language model, we want to add the type of a new subject to the cell state to replace the part of the old state that should be forgotten.

The next step is to update an old cell state. Specifically, it transforms the data from state C_{t-1} to a new state C_t. Based on the results of the previous steps, the processed signals are returned to this step for computation. Based on the outcomes of earlier steps, the processed signals will return to this step for computation, including the signal f_t corresponding to the previous forget gate. Additionally, the signal element it×C˜_t represents a newly computed value that dictates the extent of the update for each state value. The entire process is illustrated by *equation (4)* as follows:

$$C_t = f_t * C_{t-1} * \tilde{C}_t \tag{4}$$

Finally, it is necessary to determine how much of the output will be returned. The output results are based on the cell state, but in a filtered version. The process begins with the signal passing through a sigmoid layer, which determines which part of the cell state will be included in the output. Then, the cell state passes through a *tanh* function to scale the values to the range of -1 to 1 and is then multiplied by the output of a sigmoid gate, ensuring that only the part determined by the expert is returned.

$$o_t = \sigma(W_0.[h_{t-1}, x_t] + b_0) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

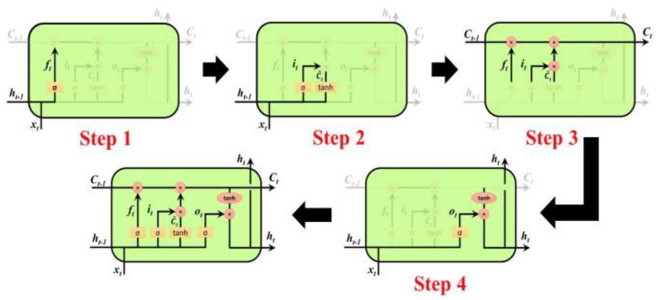

Figure 2. The neural particle model is based on the LSTM network

3.2. Gated Recurrent Unit (GRU) Network

GRU has emerged as an architecture in the RNNs. Its design addresses several limitations of traditional RNNs while providing a more compact variant of LSTM, often achieving comparable quality with significantly faster computation. The literature review [16] synthesizes various findings related to GRU, highlighting its structure, performance, and applicability across different domains. It also identifies knowledge gaps and suggests directions for future research. The structure of the GRU network is presented in *figure 3* below.

Research Article | Volume 13, Issue 2 | Pages 250-256| e-ISSN: 2347-470X

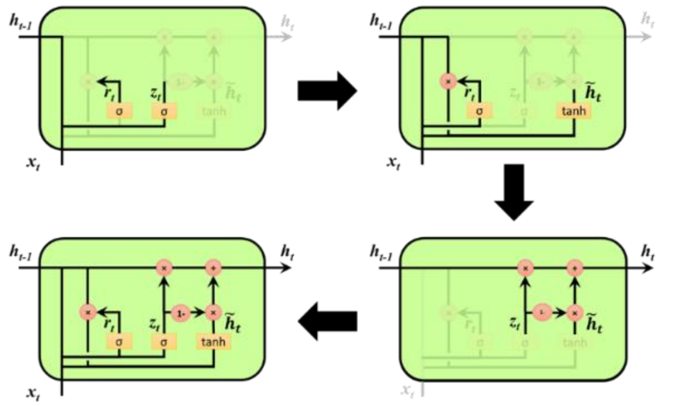

Figure 3. The operational process of a neural node in a GRU network

In *Figure 3*, the basic operation process of GRU includes the following steps:

The first step involves the reset gate and the update gate. The network is designed as vectors with elements in the range (0,1) to express convex combinations. *Equations (7)* and *(8)* are described as follows:

$$r_t = \sigma(x_t.W_{xr} + h_{t-1}.W_{hr} + b_r) \quad (7)$$

$$Z_t = \sigma(x_t.W_{xz} + h_{t-1}.W_{hz} + b_z) \quad (8)$$

The next step describes the operational state of the inputs for both the reset gate and the update gate in GRU. These inputs include the current time step input x_t and the hidden state from the previous time step h_{t-1}. The output is generated by a fully connected layer with a sigmoid activation function, as described in *equation (9)*.

$$\tilde{h}_t = \tanh(x_t.W_{xh} + (R_t * h_{t-1}) * W_{hh} + b_h) \quad (9)$$

The following process adds non-linearity through a *tanh* function to ensure that the hidden state values remain within the range (-1,1). To reduce the influence of previous states, we can multiply h_{t-1} and r_t according to each element. When the reset gate elements r_t approach1, the result resembles a standard RNN. In contrast, if the components of the reset gate rt are close to zero, the hidden state is dictated by the output of a multilayer perception, with x_t serving as its input.

Any previously existing hidden state is reset to its default value. Here, it is referred to as the potential hidden state, and it is considered "potential" because the output of the update gate still needs to be incorporated.

Finally, the effect of the update gate zt will be integrated as illustrated in the third box of *figure 3*. This gate determines the similarity between the new state h_t and the previous state h_{t-1}, as well as the extent to which the candidate hidden state h ̃_t is utilized. The gating variable z_t serves this purpose by applying a convex combination between the old and potential states. The final update equation for GRU is described in *equation (10)*.

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t \quad (10)$$

4. THEORY AND CALCULATION

4.1 The concept of data preprocessing

Data preprocessing is a crucial step in data analysis and machine learning, helping to transform raw, incomplete data into a clean, structured format. It ensures data quality by addressing inconsistencies, enhancing reliability, and preparing the data for more effective analysis. As a result, the performance of analytical algorithms and models is significantly improved.

Another key step in data preprocessing is normalization, which aims to bring all features to the same scale. For example, data can be scaled to the range [0,1] or standardized to follow a normal distribution with a mean of 0 and a standard deviation of 1.

Data preprocessing enhances data quality and ensures more accurate model performance by reducing inconsistencies. This process optimizes efficiency, resolves underlying issues, and provides a solid foundation for precise analysis, forecasting, and decision-making.

4.2. Proposed Data Preprocessing Model

The implementation process involves collecting data from the wind power plant model in Tra Vinh province. Raw data, which often contains errors, undergoes multiple preprocessing steps, including noise detection, variable filtering using correlation coefficients, and the alternating or combined application of intelligent algorithms to enhance the neural network's efficiency. Finally, a comparative analysis is conducted to identify the most suitable method for the data. *Figure 4* illustrates the proposed forecasting model. In *Figure 4*, the proposed forecasting model is carried out through the following steps:

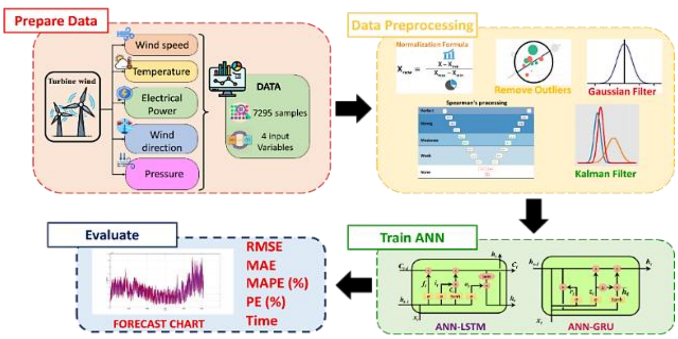

Figure 4. Proposed Forecasting Model for Wind Turbine Power Generation

Step 1: Prepare data from a real-world model or simulation to construct the training dataset. A dataset was collected from January 1 to November 16, 2022, with 24 daily samples, totaling 7,295 samples. It includes five input variables: wind speed, wind direction, temperature, pressure, and setpoint power, while the output represents generated power. Available electrical power was excluded due to its instability in wind turbine generation.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 250-256| e-ISSN: 2347-470X

Step 2: Perform data preprocessing using one or a combination of methods to clean and filter noise, such as Gaussian or Kalman filters, and to reduce data dimensionality.

Zar's paper [17] discusses the Spearman correlation coefficient, focusing on its formula, statistical significance testing, and applications. However, it does not specify a definitive threshold for deciding whether to retain or remove a variable.

However, in practice, other statistical references [18] and [19] mention widely used thresholds for assessing the strength or weakness of the Spearman correlation coefficient(ρ): Strong relationship(|ρ|>0.7), Medium relationship (0.3≤|ρ|≤0.7), Weak relationship (|ρ|<0.3).

The choice to keep or discard a variable is generally determined by statistical significance testing (ρ-value). If the ρ-value is less than 0.05, the relationship is considered statistically significant, even if the correlation coefficient is low. In summary, this paper proposes eliminating features with a coefficient smaller than 0,3. The following table presents the Spearman correlation coefficients among variables, which are explained and discussed in detail in the paragraph below.

This step involves an iterative process to generate multiple scenarios. After performing feature reduction using the Spearman correlation coefficient, as shown in Table 1, the results indicate that temperature and pressure lack correlation with actual power, with values of 0,274 and -0,119, respectively.

Table 1. Comparison results of variable importance using the Spearman correlation coefficient

	Wind speed (m/s)	Wind direction (º)	Temperature (ºC)	Pressure (mbar)	P actual (MW)
Wind speed (m/s)	1	-0,343	-0,301	0,092	-0,697
Wind direction(º)	-0,343	1	0,358	-0,452	0,345
Temperature (ºC)	-0,301	0,358	1	-0,378	0,274
Pressure (mbar)	0,092	-0,452	-0,378	1	-0,119
P actual (MW)	-0,697	0,345	0,274	-0,119	1

Step 3: Building a neural network for forecasting using LSTM and GRU models through MATLAB version 2023a on a computer with 12GB RAM and a 2.3 GHz clock speed.

Specifically, the LSTM network is configured with fundamental parameters, including three core layers [20]: an input layer, three consecutive hidden LSTM layers, and an output layer. Each LSTM layer includes 200 hidden neurons and is integrated with a Batch Normalization layer to standardize the output. After passing through the three LSTM layers, the signal goes through an additional Dropout layer with 50 neurons to mitigate overfitting. The training process is capped at 600 epochs, utilizing a step size of 1e-5 and a gradient clipping threshold of 1.

The GRU network is structured similarly to the LSTM, with the only difference being that the three hidden layers are replaced with consecutive GRU layers.

Step 4: Compare and evaluate based on accuracy and training time. Find the model that best fits the data.

5. RESULTS

The LSTM and GRU networks are trained using 12 datasets derived from a single original dataset through various preprocessing techniques. Although all 12 datasets originate from the same source, the primary objective of the study is to compare the performance of the two neural network architectures across different data transformations to identify which dataset yields the best forecasting results.
Table 2 presents the training results, showing cases 9 and 10 with low error rates. The best result is achieved using GRU with Cleaning, Normalization, and Transformation of Data (CNT) + Removing Outliers (RO) + Gaussian Filter (G) + Kalman Filter (K), yielding an RMSE of 0,006 and an MAPE of 0,516%. The figure below illustrates the forecasting results for four representative networks, showing significantly lower errors compared to training without effective data processing, with differences of ΔRMSE = 15,495 and ΔMAPE = 84,551%.

Table 2. Statistical analysis of the training errors for the LSTM and GRU neural networks across different data processing scenarios

		LSTM	**GRU**
Raw data	RMSE	**15,598**	15,555
	MAE	**11,832**	11,832
	MAPE (%)	**86,192**	85,067
	PE (%)	**96,64**	96,326
CNT	RMSE	0,171	0,172
	MAE	0,135	0,135
	MAPE (%)	11,296	11,174
	PE (%)	12,276	11,542
CNT + Reducing 1 Variable	RMSE	0,172	0,168
	MAE	0,135	0,133
	MAPE (%)	11,154	11,144
	PE (%)	12,264	11,383
CNT + Reducing 2 Variable	RMSE	0,173	0,172
	MAE	0,136	0,133
	MAPE (%)	11,231	11,018
	PE (%)	11,581	11,417
CNT+RO	RMSE	0,058	0,06
	MAE	0,038	0,038
	MAPE (%)	3,48	3,505
	PE (%)	3,58	3,626
CNT+RO+ Reducing1 Variable	RMSE	0,174	0,171
	MAE	0,138	0,135
	MAPE (%)	11,501	11,279
	PE (%)	11,789	11,557
CNT+RO Reducing2 Variables	RMSE	0,174	0,186
	MAE	0,136	0,149
	MAPE (%)	11,299	12,468
	PE (%)	11,661	12,727

CNT+RO+G	RMSE	0,056	0,055
	MAE	0,035	0,034
	MAPE (%)	3,24	3,175
	PE (%)	3,338	3,278
CNT+RO+K	RMSE	**0,006**	**0,007**
	MAE	**0,005**	**0,006**
	MAPE (%)	**0,481**	**0,516**
	PE(%)	**0,481**	**0,515**
CNT+RO+ G+K	RMSE	**0,008**	**0,006**
	MAE	**0,007**	**0,005**
	MAPE (%)	**0,651**	**0,516**
	PE(%)	**0,65**	**0,515**
CNT+RO+G+K + Reducing1 Variable	RMSE	0,081	0,084
	MAE	0,067	0,07
	MAPE (%)	5,971	6,255
	PE (%)	5,927	6,199
CNT+RO+ G+K+ Reducing2 Variables	RMSE	0,086	0,08
	MAE	0,071	0,064
	MAPE (%)	6,183	5,636
	PE (%)	6,232	5,672

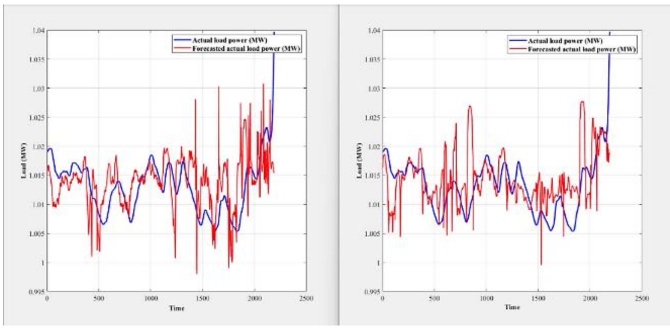

Figure 6. The forecast results of the LSTM network for processing the 9th and 10th data points

Figure 6 and *figure 7*, illustrating the power prediction graph over time from the training data set of each LSTM and GRU neural network, show that the red line (prediction) with the data "Cleaning, Normalization, and Transformation of Data + Removing Outliers + Gaussian Filter (G) + Kalman Filter" shows a relatively stable fit with the green line (target). Figure 8 shows the training runtime results for the proposed cases, consistently demonstrating the faster performance of GRU compared to LSTM.

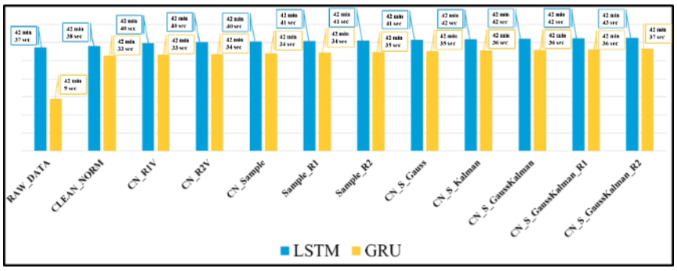

Figure 7. A chart comparing the CPU training time of LSTM and GRU

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, and the conclusions that can be drawn.

5. CONCLUSION

The results of the simulations comparing GRU and LSTM across various sizes in the proposed cases are summarized. It has been found that GRU offers advantages in speed and training performance, especially when working with smaller datasets or shorter sequences. In some instances, the GRU architecture provides a better performance-to-cost ratio, making it a viable alternative to LSTM. The GRU architecture has proven to be advantageous in a variety of applications, with data that has been "Cleaning, Normalization, and Transformation of Data + Removing Outliers + Gaussian Filter + Kalman Filter," resulting in low error, good quality, and faster training time compared to LSTM with the same 6-second dataset. The quicker processing time provides an advantage in power coordination commands and in developing discharge strategies to avoid delayed commands during discharge deadlines or when handling power system faults.

ACKNOWLEDGMENTS

This work belongs to the project grant number: T2024- 109 funded by Ho Chi Minh City University of Technology and Education, Vietnam.

REFERENCES

[1] Ghods, Ladan., & Kalantar, Mohsen. (2008). Methods for long-term electric load demand forecasting; a comprehensive investigation. 2008 IEEE International Conference on Industrial Technology, 1–4. Doi: 10.1109/icit.2008.4608469.

[2] Engineering and design hydropower proponent, "Load forecasting methods," in EM 110-2-1701, Dec 1985, appendix B

[3] D. Genethliou, and Euen A.Feinberg, "Load forecasting," Applied mathematics for restructured electric power system: optimization, control and computational intelligence (J. H. Chow, F.F. Wu, and J.J.Momoh, eds.), chapter 12, 2005, pp. 269-285.

[4] C.W.Fu, and T.T.Nguyen, "Models for long-term energy forecasting," IEEE power engineering society general meeting, vol.1, 13-17 July 2003, pp. 235-239. doi: 10.1109/PES.2003.1267174.

[5] M-F. Allawi, U-H. Abdulhameed, A. Adham, K-N. Sayl, S-O. Sulaiman, et al., "Monthly rainfall forecasting modelling based on advanced machine learning methods: tropical region as case study," Engineering Applications of Computational Fluid Mechanics, vol. 17, no. 1, August 2023, doi: 10.1080/19942060.2023.2243090.

[6] Zhao, E., Sun, S., & Wang, S. (2022). New developments in wind energy forecasting with Artificial Intelligence and big data: A scientometric insight. Data Science and Management, 5(2), 84–95. doi: 10.1016/j.dsm.2022.05.002

[7] A-T. Hoang, A-I. Olcer, H-C. Ong, W. Chen, C-T. Chong, et al., "Impacts of covid-19 pandemic on the global energy system and the shift progress to renewable energy: opportunities, challenges, and policy implications," Energy Policy, vol. 154, pp. 112322, July 2021, doi: 10.1016/j.enpol.2021.112322.

[8] S. Liu, T. Xu, X. Du, Y. Zhang, and J. Wu, "A hybrid deep learning model based on parallel architecture tcn-lstm with savitzky-golay filter for wind power prediction," Energy Conversion and Management, vol. 302, no. N/A, pp. 118122, February 2024, doi: 10.1016/j.enconman.2024.118122.

[9] Wang, Y., Chen, T., Zhou, S., Zhang, F., Zou, R., & Hu, Q. (2023). An improved wavenet network for multi-step-ahead wind energy forecasting. Energy Conversion and Management, 278, 116709. doi: 10.1016/j.enconman.2023.116709

[10] C. S. Alfredo and D-A. Adytia, "Time series forecasting of significant wave height using gru, cnn-gru, and lstm," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 6, no. 5, pp. 776–781, October 2022, doi: 10.29207/resti.v6i5.4160.

[11] F-M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: cnn, rnn, lstm, gru," arXiv.org, 2023, doi: 10.48550/arXiv.2305.17473.

[12] N. Aslam, F. Rustam, E. Lee, P-B. Washington, and I. Ashraf, "Sentiment analysis and emotion detection on cryptocurrency related tweets using ensemble lstm-gru model," IEEE Access, vol. 10, pp. 39313–39324, 2022, doi: 10.1109/ACCESS.2022.3165621.
[13] Karaman, Ö. A. (2023). Prediction of wind power with Machine Learning Models. Applied Sciences, 13(20), 11455. doi:10.3390/app132011455.
[14] Zhao, Z., Yun, S., Jia, L., Guo, J., Meng, Y., He, N., Li, X., Shi, J., & Yang, L. (2023). Hybrid VMD-CNN-GRU-based model for short-term forecasting of wind power considering spatio-temporal features. Engineering Applications of Artificial Intelligence, 121, 105982. doi:10.1016/j.engappai.2023.105982
[15] S. Ansari, T-G. Sampath Vinayak Kumar, and J. Dhillon, "Wind power forecasting using artificial neural network," 2021 4th International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE), pp. 35–37, October 2021, doi: 10.1109/RDCAPE52977.2021.9633643.
[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, doi: 10.48550/arXiv.1412.3555.
[17] J. H. Zar, "Spearman Rank Correlation," Journal of the American Statistical Association, vol. 67, no. 339, pp. 578-580, 1972
[18] D. J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 5th ed. Boca Raton, FL, USA: CRC Press, 2011.
[19] P. Schober, C. Boer, and L-A. Schwarte, "Correlation coefficients: appropriate use and interpretation," Anesthesia & Analgesia, vol. 126, no. 5, pp. 1763–1768, May 2018, doi: 10.1213/ANE.0000000000002864.
[20] Y. Yu, H. Si, X. Zeng, and Y. Ma, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," Neural Computation, vol. 31, no. 7, pp. 1235–1270, Jul. 2019. doi: 10.1162/neco_a_01199
[21] Mohd Herwan Sulaiman, Zuriani Mustaffa, solving optimal power flow problem with stochastic wind–solar–small hydro power using barnacles mating optimizer, Control Engineering Practice, Volume 106, 2021, 104672.
[22] Ahmad, Manzoor&Javaid, Nadeem &Niaz, Iftikhar& Al-Mogren, A.s &Radwan, Ayman. (2021). A Bio-inspired Heuristic Algorithm for Solving Optimal Power Flow Problem in Hybrid Power System Implementation of Published Article in IEEE Access. IEEE Access.
[23] Ugur Guvenc, Serhat Duman, Hamdi Tolga Kahraman, Sefa Aras, Mehmet Katı, Fitness–Distance Balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources, Applied Soft Computing, Volume 108, 2021, 107421.
[24] I. U. Khan, N. Javaid, K. A. A. Gamage, C. J. Taylor, S. Baig and X. Ma, "Heuristic Algorithm Based Optimal Power Flow Model Incorporating Stochastic Renewable Energy Sources," in IEEE Access, vol. 8, pp. 148622-148643, 2020, doi: 10.1109/ACCESS.2020.3015473.
[25] S. S, H. K. B, J. Reddy, R. Dash and V. Subburaj, "Dual-Topology Cross-Coupled Configuration of Switched Capacitor Converter for Wide Range of Application," 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy, 2022, pp. 796-800, doi: 10.1109/MELECON53508.2022.9843051
[26] Hemanth Kumar, B., and Makarand M. Lokhande. "Investigation of switching sequences on a generalized SVPWM algorithm for multilevel inverters." Journal of Circuits, Systems and Computers. 2019, 28, no. 02: 1950036.
[27] Karthik, N., Parvathy, A.K., Arul, R. et al. Multi-objective optimal power flow using a new heuristic optimization algorithm with the incorporation of renewable energy sources. Int J Energy Environ Eng (2021).
[28] Mouassa, S., Alateeq, A., Alassaf, A., Bayindir, R., Alsaleh, I., &Jurado, F. (2024). Optimal Power Flow Analysis with Renewable Energy Resource Uncertainty Using Dwarf Mongoose Optimizer: Case of ADRAR Isolated Electrical Network. *IEEE Access*.
[29] Kasinath Jena, Dhananjay Kumar, B. Hemanth Kumar, K. Janardhan, Arvind R. Singh, Raj Naidoo, Ramesh C. Bansal, "A Single DC Source Generalized Switched Capacitors Multilevel Inverter with Minimal Component Count", International Transactions on Electrical Energy Systems, vol. 2023, Article ID 3945160, 12 pages, 2023.
[30] Kumar, Busireddy Hemanth, Makarand Mohankumar Lokhande, Karasani Raghavendra Reddy, and Vijay Bhanuji Borghate. "An improved space vector pulse width modulation for nine-level asymmetric cascaded H-bridge three-phase inverter." Arabian Journal for Science and Engineering. 2019, 44: 2453-2465.
[31] Li, S., Gong, W., Wang, L., &Gu, Q. (2022). Multi-objective optimal power flow with stochastic wind and solar power. *Applied Soft Computing*, *114*, 108045.
[32] Kumar, Busireddy Hemanth.; and Vivekanandan Subburaj. Integration of RES with MPPT by SVPWM Scheme. Intelligent Renewable Energy Systems. 2022; 157-178. https://doi.org/10.1002/9781119786306.ch6.
[33] Huy, T. H. B., Doan, H. T., Vo, D. N., Lee, K. H., & Kim, D. (2023). Multi-objective optimal power flow of thermal-wind-solar power system using adaptive geometry estimation based multi-objective differential evolution. *Applied Soft Computing*, *149*, 110977.
[34] Pandya, S., & Jariwala, H. R. (2022). Single-and multi-objective optimal power flow with stochastic wind and solar power plants using moth flame optimization algorithm. *Smart Science*, *10*(2), 77-117.
[35] Safa Abdulwahid, Mahmoud-Reza Haghifam (2024), Short Term Load Prediction based on LSTM Network for Iraqi Thermal Power Plant. IJEER 12(4), 1461-1465. DOI: 10.37391/ijeer.120440.
[36] Duman, S., Rivera, S., Li, J., & Wu, L. (2020). Optimal power flow of power systems with controllable wind‐photovoltaic energy systems via differential evolutionary particle swarm optimization. *International Transactions on Electrical Energy Systems*, *30*(4), e12270.
[37] Karthik Nagarajan, Ayalur Krishnamoorthy Parvathy and Arul Rajagopalan, Multi-Objective Optimal Reactive Power Dispatch using Levy Interior Search Algorithm, *International Journal on Electrical Engineering and Informatics,* Volume 12, Number 3, pp.547-570, 2020.
[38] Karthik N., Parvathy A.K., Arul R., Padmanathan K. (2021) Levy Interior Search Algorithm-Based Multi-objective Optimal Reactive Power Dispatch for Voltage Stability Enhancement. In: Zhou N., Hemamalini S. (eds) *Advances in Smart Grid Technology. Lecture Notes in Electrical Engineering*, vol 688. Springer, Singapore.
[39] Zervoudakis, K., &Tsafarakis, S. (2020). A mayfly optimization algorithm. *Computers & Industrial Engineering*, *145*, 106559.
[40] Omar W. Albawab (2025), Advanced Artificial Intelligence Techniques for Fault Distance Prediction in Optical Fibres. IJEER 13(1), 89-100. DOI: 10.37391/IJEER.130113.
[41] Suy Kimsong, Horchhong Cheng, Chivon Choeung, Sophea Nam and Darith Leng (2024), Improvement of Solar Farm Performance based on Photovoltaic Modules Selection. IJEER 12(3), 951-956. DOI: 10.37391/IJEER.120328.
[42] Karthik, N., Parvathy, A. K., & Arul, R. (2019). Multi‐objective economic emission dispatch using interior search algorithm. *International Transactions on Electrical Energy Systems*, *29*(1), e2683.
[43] Karthik, N., Parvathy, A. K., & Arul, R. (2017). Non-convex economic load dispatch using cuckoo search algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, *5*(1), 48-57.