JOURNAL OF RENEWABLE ENERGY AND SMART GRID TECHNOLOGY

Vol. 20, No. 1, January-June 2025

Predicting the Temperature Trend of a Generator Using RNN, GRU, and LSTM Algorithms at Nam Ngum 1 Hydropower Plant in Laos

Bounpone Thansouphanh[1], Suttichai Premrudeepreechacharn[1*], Watcharin Srirattanachaikul[1], Kanchit Ngamsanroaj[2]

[1]Department of Electrical Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand

[2]College of Engineering and Computing, University of South Carolina, SC 29208, USA

***Corresponding author's email:** suttic@eng.cmu.ac.th

Article info:

Received: 24 December 2024
Revised: 7 February 2025
Accepted: 4 March 2025
DOI:
10.69650/rast.2025.260236

Keywords:

Recurrent Neural Network
Long Short-Term Memory
Gated Recurrent Unit
Predictive Maintenance
Generator's Stator Winding

ABSTRACT

Synchronous generators are integral to the operation of hydropower plants. Stator faults, including short circuits, open circuits, and inter-turn faults, can cause severe performance issues and even catastrophic failures if not identified and mitigated promptly. Traditional generator monitoring methods such as periodic inspections and time-based maintenance often fail to detect subtle or evolving faults. This study proposes an advanced predictive maintenance approach utilizing deep learning techniques to monitor generator health at Nam Ngum-1 (NNG-1) Hydropower Plant in the Lao People's Democratic Republic (PDR). Accurate temperature forecasting is vital for predictive maintenance, as excessive heat can lead to performance degradation and costly downtime. Using time-series data from the plant's supervisory control and data acquisition (SCADA) system, including parameters such as power output, voltage, current, and cooling system temperatures, this research evaluates the models' ability to capture temporal dependencies critical for precise trend prediction. Among the models tested, the results demonstrate that LSTM, with one hidden layer, achieved the highest accuracy based on MSE, RMSE, MAE, and R-squared, outperforming GRU, which had an R-squared of 98.60%, and RNN, which achieved 97.04%. When implemented with two hidden layers, LSTM maintained its superior performance with an R-squared of 98.34%, compared to GRU's 97.93% and RNN's 92.68%. These results demonstrate LSTM's exceptional capability in capturing both short- and long-term temperature dependencies, making it particularly suitable for predictive maintenance applications. The model's high accuracy in temperature forecasting enables early fault detection, helping to prevent performance degradation and reduce costly downtime in hydropower operations.

1. Introduction

The Electricité du Laos Generation Public Company (EDL-Gen), a state-owned enterprise, oversees the operation of 13 hydropower plants along with additional electricity generation facilities. Among these, the Nam Ngum 1 (NNG-1) Hydropower Plant [1] stands as the largest facility managed by EDL-Gen [1] situated approximately 90 kilometers north of Vientiane, the capital city of the Lao People's Democratic Republic (PDR), as depicted in Figure 1, NNG-1 is strategically connected to the Mekong River, a vital waterway in the region. The Mekong supports electricity production at various sites, collectively capable of generating up to 1,600 MW. The NNG-1 plant itself houses eight synchronous generators, consisting of: Two smaller units each rated at 17.5 MW. Six larger units each rated at 40 MW. This configuration enables NNG-1 to deliver a total installed capacity of 275 MW.

Synchronous generators are pivotal in hydropower plants, converting mechanical energy into electrical electricity [2]. However, because of aging, constant use, load variations, temperature changes, and mechanical stress, these generators are prone to wear and degradation overtime. Monitoring the stator winding temperature is crucial, as excessive heat can compromise insulation, shorten the generator's lifespan, and lead to costly unplanned outages. Predicting temperature trends allows for proactive maintenance strategies, reducing operational risks and costs.

Numerous studies have explored temperature prediction methods in various applications. For instance, Y Li et al. [3] proposed a condition-based maintenance (CBM) strategy that considers imperfect maintenance, where equipment is only partially restored after repair, Yu Yi et al. [4] proposed a wellbore temperature model for shallow drilling and pump shutdown cycles based on thermal principles. Similarly, Q. Zhang et al. [5] introduced a data-driven approach employing dynamic principal component analysis and an extreme learning machine to forecast calciner outlet temperatures in cement production.

In the energy domain, M. Abumohsen et al.[6] forecasted electrical loads in Palestine using deep learning models (LSTM, GRU, and RNN), increasing dependability and cost effectiveness.

N.Klungsida and P. Maneechot et al. [7] used similar algorithms to predict energy consumption patterns for a solar-powered EV charging station in Thailand. L. Wensheng et al. [8] developed a regional energy load forecasting method integrating Convolutional Neural Network (CNN) and LSTM for enhanced accuracy. A.Badjan et al. [9] combined seasonal and trend decomposition with Loess (STL) and LSTM networks to predict energy consumption in base stations.

In high-performance applications, C. Sciascera et al. [10] highlighted the importance of accurate thermal modeling of electrical machines, proposing analytical models to enhance computational efficiency for aerospace systems. Other researchers [11] have advanced multi-parameter optimization models for thermoelectric generator prediction, further demonstrating the efficacy of deep learning algorithms for time-series temperature data analysis.

These studies underscore the potential of advanced algorithms like LSTM, GRU, and RNN to accurately forecast temperature trends, enable better maintenance strategies, and enhance the operational reliability of critical systems like hydropower plants.

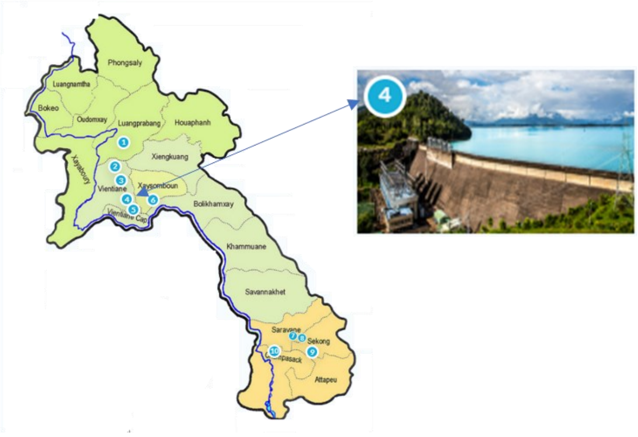

Fig. 1 The Nam Ngum 1 Hydropower Plant's location.

Predictive maintenance for generators has a number of important benefits. First, it increases the reliability and availability of electrical power, this is necessary to guarantee the electrical grid stability. Second, it lowers upkeep expenses by doing away with pointless processes and expensive yearly inspections. Last, it aids in avoiding unforeseen failures, increases industrial processes' productivity, and minimizes financial losses caused by production disruptions.

Traditionally, synchronous generator maintenance relies on periodic inspections and time-based strategies, frequently leading to either excessive or insufficient upkeep. These inefficiencies raise operating expenses and fail to take into consideration the dynamic, situations that generators face in real time while they are operating. Consequently, there is a growing need to develop effective and efficient methods for synchronous generator maintenance and monitoring, especially in industrial applications and power generation.

The principal contribution of this study is the development of predictive models for the temperature trends of generator stator windings utilizing deep learning techniques, namely Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs), and Long Short-Term Memory (LSTM) networks. The collection consists of data from the SCADA system of the Nam Ngum-1 Hydropower Plant as well as records of preventive or time-based maintenance (TBM). derived from annual inspections. Deep learning and statistical analysis are applied to process this data, enabling the prediction of the temperature trend and optimizing maintenance schedules.

This section introduces the concept of temperature trend forecasting. The structure of the paper is as follows: An outline of the deep learning algorithms is given in section 2, the process for creating models with RNN, LSTM, and GRU is described in section 3, the experimental results are shown in section 4, and important conclusions and ramifications are discussed in section 5.

2. Deep Learning Algorithms

This section provides an overview of three deep learning algorithms: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) [12]. These algorithms were selected due to their extensive usage in forecasting jobs and demonstrated performance. Despite having identical goals, all deep learning algorithms use different mathematical frameworks, each with its own advantages and disadvantages. By examining the correlations between variables throughout the learning process, these techniques forecast dependent variable values from independent variables. This study forecasts a generator's temperature trends using RNN, LSTM, and GRU.

2.1 Recurrent Neural Networks (RNNs)

RNNs preserve and spread information over time steps in order to process sequential data. Unlike typical neural networks, RNNs incorporate a feedback loop that allows them to store information from prior time steps, enabling them to simulate temporal dependencies effectively. By capturing contextual information from previous inputs, this feedback mechanism generates a hidden state that functions as a memory, RNNs are ideally suited for tasks that require comprehension of temporal linkages and sequential patterns by utilizing this memory.

Figure 2 shows both folded and unfolded schematics to illustrate how RNNs works. Each time step in these diagrams denotes a distinct point in the sequence, and the arrows show how the data moves from one step to the next. At each step, the model receives two inputs: At time step *t*, the input data, denoted as x_t , is processed alongside the hidden state h_{t-1} which encapsulates information from the previous time step. The output corresponds to the updated hidden state at time *t*, denoted by h_t .

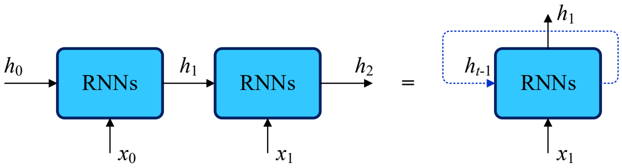

Fig. 2 Diagrammatic depiction of RNNs folded and unfolded.

Figure 3 depicts the RNNs architecture that Elman (1990) first proposed. Furthermore, equations (1) and (2) represent the hidden states.

$$h_t = \tanh\,(w_x x_t + w_h h_{t-1} + b_h) \tag{1}$$

$$o_t = w_o h_t + b_o \tag{2}$$

where $\tanh(a) = (e^{a} - e^{-a}) / (e^{a} + e^{-a})$ is the hyperbolic tangent function, $x_t \in \mathbf{R}^m$, and $h_{t-1}\varepsilon \in \mathbf{R}^m$ represents the input vectors, while $o_t \in \mathbf{R}^m$, and $h_t \in \mathbf{R}^m$ represents the output vectors. Furthermore, the bias vectors are called to as $b_h \in \mathbf{R}^m$, and $b_o \in \mathbf{R}^m$, and symbols are used to represent the weight matrices for the input-to-hidden, hidden-to-hidden, and hidden-to-output connections, respectively. $w_x \in \mathbf{R}^{m\times m}$, $w_h \in \mathbf{R}^{m\times m}$, and $w_o \in \mathbf{R}^{m\times m}$.

Nevertheless, the vanishing gradient problem is a significant challenge that RNNs may encounter. When the gradients that update the network weights get too little, this issue occurs. Results in training the model ineffectively. Therefore, it can be difficult for the model to capture long-term dependencies in data, this is typical of time series in finance (Bengio et al, 1994). More sophisticated models like LSTM and GRU have been developed as a result of this constraint.

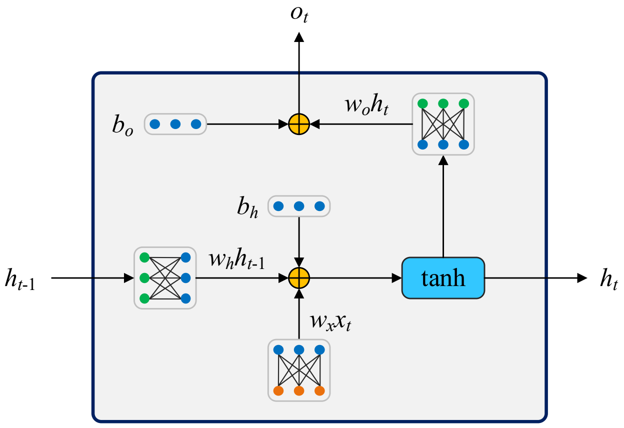

Fig. 3 Architecture of Recurrent Neural Networks.

2.2 Long Short-Term Memory (LSTM)

To address the problem of disappearing gradients in RNNs, LSTM was suggested (Hochreiter and Schmid-Huber, 1997). One of the advantages of LSTM is the incorporation of a cell state, $c_{t\text{-}1}$, This allows long-term dependencies between sequences to be captured by the model. As shown in Fig.4 the LSTM architecture consists of three gates: the forget gate, the input gate, and the output gate . These gates control the information flow and update the cell state, allowing the model to retain or discard information as necessary.

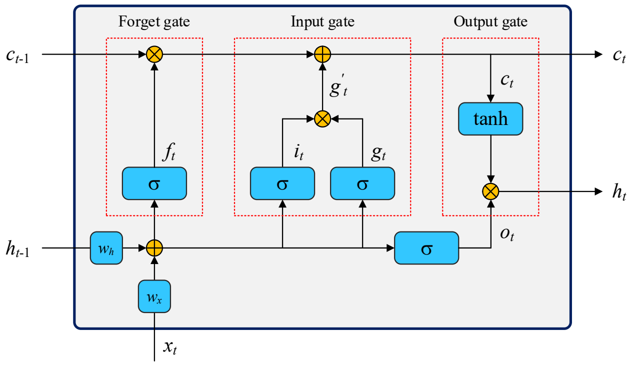

Fig. 4 Architecture of Long Short-term Memory.

The forget gate $f_t \in \mathbf{R}^m$ determines which information from the previous cell state needs to be forgotten and can be computed as:

$$f_t = \sigma(w_{hf} h_{t-1} + w_{xf} x_t + b_{hf} + b_{xf}) \tag{3}$$

$$\tilde{c}_t = f_t \odot c_{t-1} \tag{4}$$

where the definition of the sigmoid function, or activation function α, is $\sigma(\alpha) = 1/(1 + e^{-\alpha})$. The input vectors are $x_t \in \mathbf{R}^m$, and $h_{t-1} \in \mathbf{R}^m$, weight matrices $w_{hf} \in \mathbf{R}^{m\times m}$ and $w_{xf} \in \mathbf{R}^{m\times m}$ for bias vectors and the forget gate $b_{xf} \in \mathbf{R}^m$ and $b_{hf} \in \mathbf{R}^m$ in line with the forgotten gate. The prior cell state is indicated $c_{t-1} \in \mathbf{R}^m$, and the candidate cell state is represented by $\tilde{c}_t \in \mathbf{R}^m$. The symbol $\odot$ signifies an element-wise multiplication operation.

Which additional information should be added to the cell state is decided by the input gate $\bar{g} \in \mathbf{R}^m$ The input gate calculation can be carried out in this way:

$$i_t = \sigma(w_{hi} h_{t-1} + w_{xi} x_t + b_{hi} + b_{xi}) \tag{5}$$

$$g_t = \tanh(w_{hi} h_{t-1} + w_{xi} x_t + b_{hi} + b_{xi}) \tag{6}$$

$$\bar{g} = g_t \otimes i_t \tag{7}$$

$$c_t = \tilde{c}_t \otimes \bar{g}_t \tag{8}$$

where the activation functions used are σ and tanh. The input vectors are $x_t \in \mathbf{R}^m$ and $h_{t-1} \in \mathbf{R}^m$ indicate the current input and the prior hidden state, respectively. Weight matrices $w_{ho} \in \mathbf{R}^{m\times m}$ and $w_{xo} \in \mathbf{R}^{m\times m}$ the output gate uses them, while $b_{xo} \in \mathbf{R}^m$ and $b_{ho} \in \mathbf{R}^m$ are the bias vectors that correspond to it. The cell state is also taken into account while calculating the output gate $c_t \in \mathbf{R}^m$ Finally, the vector of output $h_t \in \mathbf{R}^m$ at time t is produced by integrating these components with the help of relevant mathematical operations and activation functions.

2.3 Gated Recurrent Unit (GRU)

Another sophisticated RNN design that addresses RNN drawbacks including the vanishing gradient issue is GRU. Compared to LSTM, it has a simpler structure and fewer gating mechanisms. As illustrated in Figure 5, the Gated Recurrent Unit (GRU) architecture incorporates an update gate and a reset gate, which control the flow of information within the network. GRU can avoid the vanishing gradient issue while still identifying and remembering significant patterns over lengthy periods by selectively updatingand reseting the hidden state.

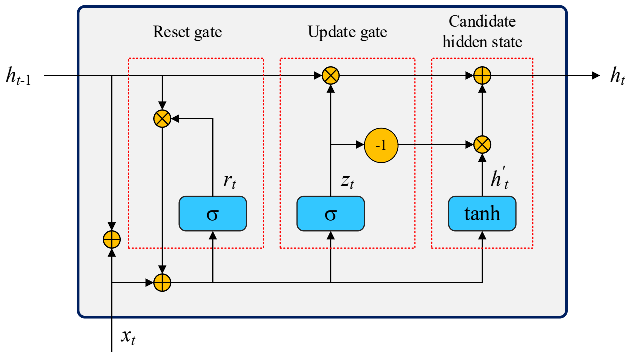

Fig. 5 Architecture of Gated Recurrent Unit. The amount of the prior.

hidden state that should be forgotten or reset is decide by the reset gate $r_t \in \mathbf{R}^m$ and can be computed as follows:

$$r_t = \sigma(w_{hr}h_{t-1} + w_{xr}x_t + b_r + b_{xr}) \qquad (9)$$

The purpose of activation is σ, the input vectors are shown as $x_t \in \mathbf{R}^m$ and $h_{t-1} \in \mathbf{R}^m$, where x_t is the current input and h_{t-1} is the prior hidden state. For the reset gate, weight matrices $w_{hr} \in \mathbf{R}^{m\times m}$ and $w_{xr} \in \mathbf{R}^{m\times m}$ are employed, in addition to bias vectors $b_{xr} \in \mathbf{R}^m$ and $b_{hr} \in \mathbf{R}^m$.

The portal of updates $z_t \in \mathbf{R}^m$ GRU architecture determines the extent to which the previous hidden state and the new candidate hidden state are combined. The following formula could be used to determine the update gate z_t .

$$z_t = \sigma(w_{hz}h_{t-1} + w_{xz}x_t + b_{hz} + b_{xz}) \qquad (10)$$

where the function of activation is σ, $x_t \in \mathbf{R}^m$ and $h_{t-1} \in \mathbf{R}^m$ are input vectors, $w_{hz} \in \mathbf{R}^{m\times m}$ and $w_{xz} \in \mathbf{R}^m$ are weight matrices, while $b_{xz} \in \mathbf{R}^m$ and $b_{hz} \in \mathbf{R}^m$ stand for bias vectors for the update gate.

The amount of new data that is added to the existing hidden state in order to update it is determined by the candidate hidden state and can be computed as follows:

$$\bar{h}_t = \tanh(w_{h\bar{h}}(r_t \odot h_{t-1}) + w_{x\bar{h}}x_t + b_{h\bar{h}} + b_{x\bar{h}}) \qquad (11)$$

$$h_t = (z_t \odot h_{t-1}) + \left((1 - z_t) \odot \bar{h}_t\right) \qquad (12)$$

where the function of activation is $\tanh$, $x_t \in \mathbf{R}^m$ and $h_{t-1} \in \mathbf{R}^m$ are vectors used as inputs, $w_{h\bar{h}} \in \mathbf{R}^{m\times m}$ and $w_{x\bar{h}} \in \mathbf{R}^{m\times m}$ are matrices of weight, while $\bar{b}_{hh} \in \mathbf{R}^m$ and $b_{x\bar{h}} \in \mathbf{R}^m$ stand for bias vectors, $\bar{h}_t \in \mathbf{R}^m$ is the concealed condition of the candidate, and $h_t \in \mathbf{R}^m$ is the state that is concealed.

2.4 Selection of Metrics

In regression analysis, various statistical metrics are utilized to assess model performance [13]. This study focuses primarily on Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R²) for evaluating deep learning models. These metrics serve as the basis for testing the models and selecting the optimal ones according to the performance criteria outlined below:

1) The Mean Squared Error (MSE) quantifies the average of the squared differences between the actual and predicted values. The formulation for calculating MSE is presented in Equation (13).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(|yt - yt^p|) \qquad (13)$$

where yt represents the actual data value, and yt^P is the predicted data value.

2) The Mean Absolute Error (MAE) calculates the average magnitude of the absolute differences between observed and predicted values. The computation of the Root Mean Squared Error (RMSE) is presented in Equation (14).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|yt - yt^p|)^2} \qquad (14)$$

3) The Mean Absolute Error (MAE) quantifies the mean of the absolute differences between actual observations and predicted values. The formula for computing MAE is presented in Equation (15).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|) \qquad (15)$$

4) The coefficient of determination (R²) is a statistical measure ranging from 0 to 1 that reflects the predictive accuracy of a model. The method for calculating R² is provided in Equation (16).

The following equation is used to determine the coefficient of determination (R^2):

$$R^2 = 1 - \frac{SS_{regression}}{SS_{total}} \qquad (16)$$

Where regression $SS_{regression}$ the explained sum of squares, another name for the regression sums of squares, and SS_{total}The total sum of all squares.

3. Methodology

This section describes the methodology employed to predict short-term temperature trends through deep learning techniques, as depicted in Figure 6. The procedure begins with the compilation of the dataset, followed by detailed preprocessing steps, including data normalization and the selection of significant features [7]. Subsequently, deep learning models—namely Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs)—are constructed and trained on historical operational data to forecast temperature patterns. Model optimization is performed by fine-tuning key hyperparameters, including the optimizer, activation functions, learning rate, batch size, number of epochs, and the number of hidden layers. Model performance is evaluated using appropriate metrics, and the model achieving the highest predictive accuracy is selected. All experiments were implemented in Python 3, utilizing libraries such as Scikit-learn, Pandas, Seaborn, Matplotlib, TensorFlow, and NumPy. The computational processes were executed on a machine featuring an 11th Gen Intel® Core™ i7-1165G7 processor operating at 2.80 GHz, equipped with 16 GB RAM, and running Windows 10.

❖ *Data Collection and Preprocessing*

This study utilized a time-series dataset recorded at 15-minute intervals over a one-year timeframe, from October 2023 to September 2024, comprising 34,828 data points. The dataset features seven primary variables: timestamp, active power output (MW), generator voltage (V), generator current (A), cooling system outlet temperature (°C), cooling system inlet temperature (°C), and the mean stator winding temperature (°C). These data were directly extracted from the Supervisory Control and Data Acquisition (SCADA) system of the Nam Ngum 1 Hydropower Plant.

The data preparation process began with a thorough cleaning phase to eliminate missing or irrelevant entries. Subsequently, we conducted feature extraction, applied data scaling, and divided the dataset to facilitate modeling and analysis. For normalization, the min-max scaling technique was employed, which adjusted all numerical values within the dataset to fall between 0 and 1. This normalization process is essential to prevent any single variable from disproportionately influencing the model due to its scale. The mathematical expression used for this scaling is presented in Equation (17).

$$\bar{X} = X - MinMax - Min \tag{17}$$

where the variable X refers to the original unnormalized input variables, Min denotes the minimum value within the dataset, Max corresponds to the maximum value in the dataset, and $\bar{X}$ indicates the normalized input variable. In the concluding phase of data preprocessing, the dataset was divided into three non-overlapping subsets: 70% for training, 15% for validation, and 15% for testing. The training set facilitated the model's learning of intrinsic patterns and correlations within the historical data. The validation set served for hyperparameter adjustment and performance fine-tuning, using data not seen during training. The test set, withheld until the final stage, ensured an objective and unbiased evaluation of the model's accuracy and its ability to generalize to real-world, unseen data.

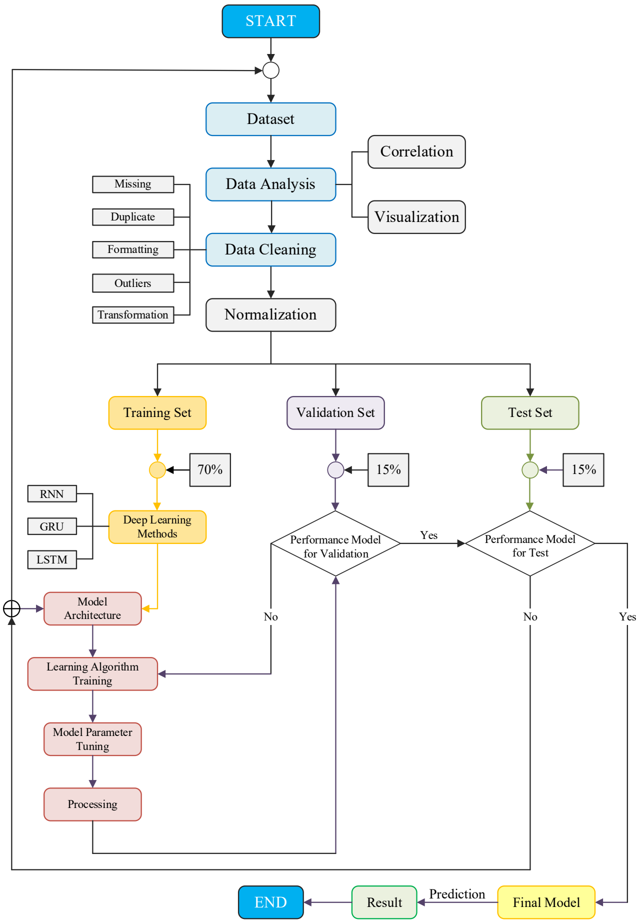

Fig. 6 Deep Learning Prediction Model Process.

4. Experimental and Results

Performance metrics are utilized to evaluate the efficiency of each model following the deployment of the RNN, GRU, and LSTM models. The expected outcomes of each method are shown in this parts, along with performance metrics for each model that were determined using MSE, R-squared, RMSE, and MAE.

Optimization strategies were systematically applied to each model to enhance prediction accuracy. Key configuration parameters included two hidden layers, a batch size of 32, a learning rate of 0.001, the use of the Adam optimizer, and the hyperbolic tangent activation function. The RNN, LSTM, and GRU models were trained on datasets partitioned into 70% for training, 15% for validation, and 15% for testing. This distribution was chosen to ensure a comprehensive learning process, effective hyperparameter tuning, and a rigorous evaluation of generalization performance. The training subset enabled the models to learn temporal patterns, the validation subset was employed to optimize hyperparameters and mitigate overfitting, and the test subset provided an unbiased assessment on unseen data. This splitting strategy was considered appropriate for this study, balancing training efficiency with evaluation reliability. Each model was consistently trained using the Adam optimizer and a fixed learning rate, while the influence of different numbers of hidden layers will be elaborated in the subsequent section.

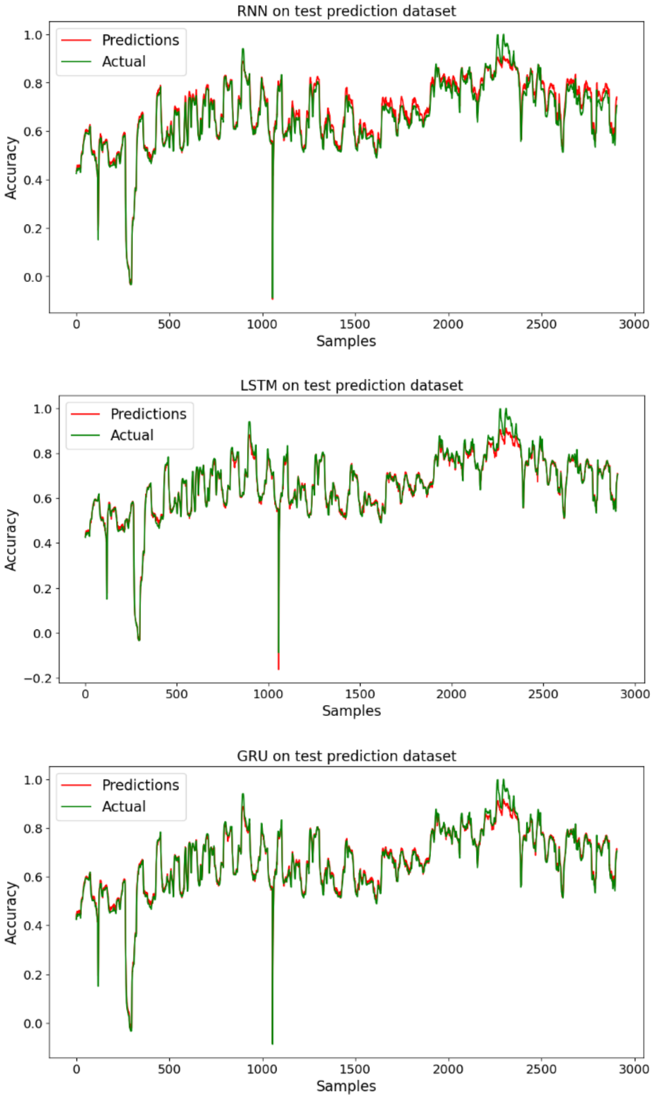

Fig. 7 Temperature prediction patterns produced by each model utilizing the Adam optimizer with a configuration of one hidden layer.

Figure 7 illustrates the comparison between actual and predicted temperature values, depicted in green and red, respectively, for the RNN, LSTM, and GRU models. All models were evaluated using a fixed learning rate of 0.001. The LSTM model demonstrated the highest level of predictive accuracy, recording an R-squared (R²) value of 98.70% and exhibiting minimal errors, particularly around peak temperature points. Similarly, the GRU model achieved an R² value of 98.60%, providing predictive results comparable to those of the LSTM. In comparison, the RNN model attained an R² of 97.04%, reflecting adequate predictive performance, though slightly lower than that of the LSTM and GRU.

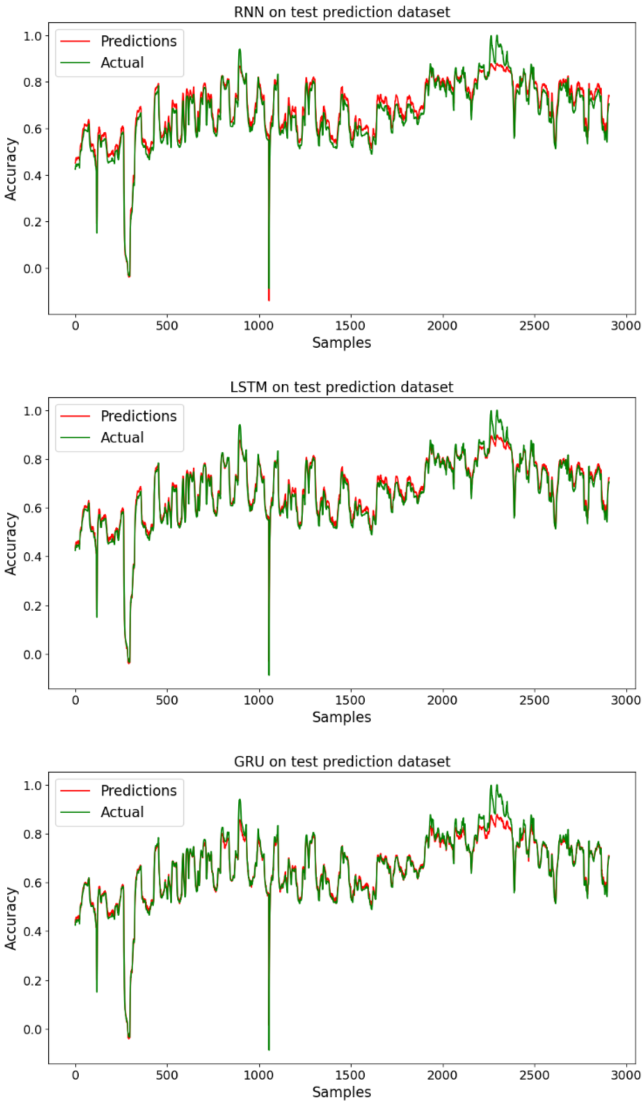

Fig. 8 Temperature prediction patterns produced by each model utilizing the Adam optimizer with a configuration of two hidden layers.

Figure 8 depicts the comparison between observed and predicted temperature trends for the RNN, LSTM, and GRU models. All models were evaluated with a learning rate set at 0.001. The LSTM model outperformed the others, attaining an R-squared (R²) value of 98.34% and achieving the lowest error rate. The GRU model delivered its optimal performance with an R² value of 97.93%, while the RNN model recorded its best result at an R² value of 92.68%. These results confirm that the LSTM model provided the highest prediction accuracy among the models evaluated.

Table 1 presents the testing outcomes for the RNN, LSTM, and GRU models, evaluated under different hyperparameter settings, including learning rates and the number of hidden layers. The highest predictive performance for all models was observed with one hidden layer and a learning rate of 0.001. Under these settings, the LSTM model attained an R-squared (R²) value of 98.70%, followed by the GRU model with an R² of 98.60%, and the RNN model with an R² of 97.04%.

Optimal results were obtained by configuring the LSTM model with two hidden layers, after applying the Adam optimization algorithm across the RNN, LSTM, and GRU models in two different architectural setups: one and two hidden layers. In this configuration, the model achieved an MSE of 0.0003, an R-squared (R²) value of 98.34%, an RMSE of 0.0181, and an MAE of 0.0137.

5. Conclusions

The objective of this study was to design and evaluate RNN, GRU, and LSTM models for predicting stator winding temperature trends at the Nam Ngum 1 Hydropower Plant in Laos. Among the models, the LSTM with a single hidden layer yielded the highest predictive accuracy, recording an MSE of 0.0003, RMSE of 0.0161, MAE of 0.0114, and an R² of 98.70%, outperforming the GRU (98.60%) and RNN (97.04%) counterparts. With two hidden layers, the LSTM further improved its performance, achieving an R² of 98.34%, exceeding the results of the GRU (97.93%) and RNN (92.68%). These results highlight the superior capability of the LSTM model to capture complex temporal dependencies, both short- and long-term, within the dataset. Its sophisticated memory architecture makes it a powerful tool for predictive maintenance strategies, enhancing reliability, minimizing operational downtime, and optimizing maintenance cycles. The findings affirm the potential of LSTM-based predictive models to drive greater operational efficiency and sustainability in hydropower plant management.

Future research directions should focus on enhancing both the technical capabilities and practical implementation of predictive maintenance models. Key areas for advancement include data enrichment by integrating additional variables such as environmental conditions, dynamic load patterns, and vibration signatures. To address current limitations in data availability, researchers could explore techniques for synthetic data generation and advanced oversampling methods to improve model robustness and generalization. The implementation scope could be expanded through real-time deployment of LSTM models with particular emphasis on developing multi-step forecasting capabilities for more comprehensive maintenance planning. The incorporation of advanced interpretability methods, including Shapley Additive Explanations (SHAP) and attention mechanisms, has the potential to significantly improve model transparency, thereby equipping maintenance teams with clearer, actionable insights to facilitate more informed decision-making processes. To ensure broader applicability and system reliability, future work should investigate hybrid modeling approaches, develop sophisticated methods for handling missing SCADA data, and validate model performance across multiple hydropower facilities. Additionally, research should focus on seamless integration with existing energy management systems and conduct detailed economic feasibility analyses to quantify the cost-benefit ratio of implementing these advanced predictive maintenance solutions

Table 1 Results from Adam Optimizer for each model.

Layer	Learning rate	Model	MSE	R-Squared	RMSE	MAE
One Hidden Layer	0.001	RNN	0.0006	97.04%	0.0242	0.0210
	0.001	LSTM	0.0003	98.70%	0.0161	0.0114
	0.001	GRU	0.0003	98.60%	0.0167	0.0126
Two Hidden Layers	0.001	RNN	0.0014	92.68%	0.0380	0.0341
	0.001	LSTM	0.0003	98.34%	0.0181	0.0122
	0.001	GRU	0.0004	97.93%	0.0202	0.0137

Acknowledgments

The authors gratefully acknowledge the support received from Electricité du Laos (EDL) and the Electricity Generating Authority of Thailand (EGAT). They also wish to extend special appreciation to the Faculty of Engineering, Chiang Mai University, for the financial support provided for this research.

References

[1] EDL-Gen. WOLLY OWN BUSINESS, <https://edlgen.com.la/hpp-wholly-owned-business> (2010).

[2] Verginadis, D., Antonino-Daviu, V., Karlis, A. and Danikas, M. Diagnosis of stator faults in synchronous generators: Short review and practical case. in *International Conference on Electrical Machines (ICEM)*. (2020), 2381-4802, doi: https://doi.org/10.1109/ICEM49940.2020.9270936.

[3] Li, Y., Han, X., Xu, B. and Wang, Y. A condition-based maintenance approach to an optimal maintenance strategy considering equipment imperfect maintenance model. in *2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*. (2015) 1466-1471, doi: https://doi.org/10.1109/DRPT.2015.7432469.

[4] Yi, Y., Xuerui, W. and Ke, K., Prediction model and distribution law study of temperature and pressure of the wellbore in drilling in Arctic region, *Petroleum Drilling Techniques.* 49(3) (2021) 11-20*, doi:* https://dx.doi.org/10.11911/syztjs.2021047.

[5] Zhang, Q., Jiao, T., Ma, Y., Zhang, Y., Wu, X. and Wu, P. Temperature prediction of generator carbon brush based on LSTM neural network. in *2021 China Automation Congress (CAC)*. (2021), 5050-5055, doi: https://doi.org/10.1109/CAC53003.2021.9728612.

[6] Abumohsen, M., Owda, A., Y. and Owda, M., Electrical load forecasting using LSTM, GRU, and RNN algorithms. *Energies.* 16 (2023) 2283, doi: https://doi.org/10.3390/en16052283.

[7] Klungsida, N., Maneechot, P., Butploy, N. and Khiewwan, K., Forecasting Energy Consumption from EV Station Charging Using RNN, LSTM, and GRU Neural Network. *Journal of Renewable Energy and Smart Grid Technology.* 19(1) (2024) 1-6, doi: https://doi.org/10.69650/rast.2024.254636.

[8] Wensheng, L., Kuihua, W., Liang, F., Hao, L., Yanshuo, W. and Can, C. A region-level integrated energy load forecasting method based on CNN-LSTM model with user energy label differentiation. in *2020 5th International Conference on Power and Renewable Energy (ICPRE)*. (2020), 154-159, doi: https://doi.org/10.1109/ICPRE51194.2020.9233226.

[9] Badjan, A., Rashed, G., I., Gony, H. A. I., Bahageel, A. O. M., Hualiang, F. and Shaheen, H., I. Predictive Modelling of Base Station Energy Consumption: Insights and Application of STL-LSTM Networks. in *2024 9th Asia Conference on Power and Electrical Engineering (ACPEE)*. (2024), 2499-2503, doi: https://doi.org/10.1109/ACPEE60788.2024.10532372.

[10] Sciascera, C., Giangrande, P., Papini, L., Gerada, C. and Galea, M., Analytical thermal model for fast stator winding temperature prediction. *IEEE Transactions on Industrial Electronics.* 64(8) (2017) 6116-6126, doi: https://doi.org/10.1109/TIE.2017.2682010.

[11] Qing, S., Rezaniakolaei, A., Rosendahl, L. A. and Gou, X., An analytical model for performance optimization of thermoelectric generator with temperature-dependent materials. *IEEEE Access.* 6 (2018) 60852-60861, doi: https://doi.org/10.1109/ACCESS.2018.2874947

[12] Safari, M., Nakharutai, N., Chiawkhun, P. and Phetpradap, P. DEA-RNNs: An Ensemble Approach for Portfolio Selection in the Thailand Stock Market. in *Partial Identification in Econometrics and Related Topics*. (2024), 453-467, doi: https://doi.org/10.1007/978-3-031-59110-5_30.

[13] Plevris, V., Solorzano, G., Bakas, N. and Ben Seghier, M. Investigation of performance metrics in regression analysis and machine learning-based prediction models. in the *8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2022)*. (2022), 1-25, doi: https://doi.org/10.23967/eccomas.2022.155.