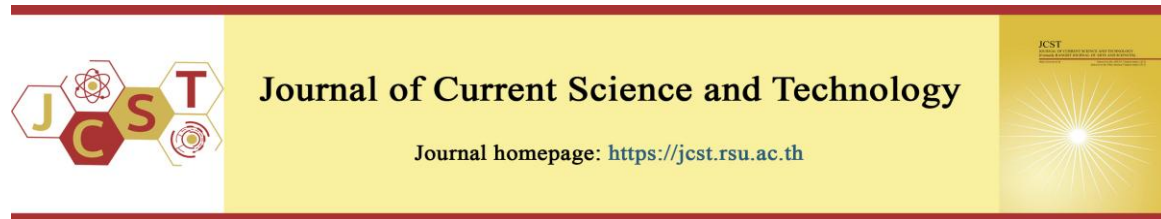


Cite this article: Deevijit, C., Kiatsiriroat, T., Deethayat, T., & Asanakham, A. (2025). Deep learning approach for predicting thermal behavior of hydropower generator-stator: A case study of a hydropower power plant in Thailand. *Journal of Current Science and Technology*, 15(4), Article 145. <https://doi.org/10.59796/jcst.V15N4.2025.145>



Deep Learning Approach for Predicting Thermal Behavior of Hydropower Generator-Stator: A Case Study of a Hydropower Power Plant in Thailand

Chinachote Deevijit¹, Tanongkiat Kiatsiriroat², Thoranis Deethayat², and Attakorn Asanakham^{2,*}

¹Energy Engineering Program, Faculty of Engineering and Graduate School, Chiang Mai University, Chiang Mai 50200, Thailand

²Thermal System Research Laboratory, Department of Mechanical Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand

*Corresponding author; E-mail: attakorn.asana@cmu.ac.th

Received 11 June 2025; Revised 14 July 2025; Accepted 23 July 2025; Published online 20 September 2025

Abstract

Hydropower generation is a cost-effective and environmentally friendly energy source that converts the kinetic energy of flowing water into electricity. However, temperature control in power generators, particularly in the conductor windings in the stator, remains a significant challenge for maintaining power generation performance. Several factors influence temperature, and their relationships are quite complex, making it difficult to solve the problem using standard theoretical approaches. This research developed a deep learning model to monitor temperature trends in the conductor windings of a 125 MW hydropower plant in Thailand. Data collected between 2018 and 2021 on electricity generation, reservoir water levels, water and air flow rates, inlet temperatures at the heat exchanger, and conductor winding temperatures were used to train and validate the models. The study implemented three neural network models: a Feedforward Neural Network (FNN), a Multilayer Feedforward Neural Network (MFNN), and a Long Short-Term Memory (LSTM) network. The results showed that the LSTM model provided the most accurate predictions, with a Mean Squared Error (MSE) of 0.00373. Shapley Additive exPlanations (SHAP) values were used to interpret the model predictions, identifying key variables such as electricity generation, water temperature, and water flow rate as the most influential factors affecting system behavior. The findings suggest that deep learning models can effectively predict temperature variations, enabling proactive maintenance and improving operational efficiency in hydropower plants.

Keywords: *deep learning; thermal behavior prediction; hydropower generator-stator*

1. Introduction

Renewable energy is derived from natural sources, such as solar, wind, hydropower, geothermal, biomass, and biogas (Electricity Generating Authority of Thailand, n.d.). Hydropower is commonly regarded as a significant energy source for generating electricity due to its origin from natural sources and the lack of harmful waste production for the environment. Hydropower generation converts the kinetic energy of moving water into electricity using turbines and generators (Aarons et al., 2015). In electricity generation, the rotor generates a rotating magnetic field within the stator windings, inducing an

electromotive force according to Faraday's law. During this process, copper and core losses occur, which become heat losses in the generator, an unavoidable factor. Controlling the temperature of the stator windings is crucial (Desingu et al., 2018), as inadequate temperature control can result in emergency shutdowns and possible damages.

Typically, heat exchangers are installed around the stator windings of the hydropower generator to transfer heat between the warm air in the stator windings chamber and the cool water in the heat exchanger, which uses water from a reservoir, as shown in Figure 1. The temperature inside the stator

windings chamber of a hydropower generator can rise due to various factors, including a sudden increase in power generation load and fouling from the cooling water in the heat exchanger (Khun et al., 2025). Generally, the performance of the heat exchanger is assessed through the UA coefficient or heat exchanger effectiveness (ϵ), and if there is a significant decrease in these values, maintenance and cleaning of the heat exchanger are required (Asvapoositkul & Kuansathan, 2016; Nogueira & Nogueira, 2022). Additionally, fluctuations in water flow rates, changes in water or air temperature, local weather conditions, and reservoir water level (Cui et al., 2021; Doost & Majlessi, 2015; Navarro & Cabezas-Gómez, 2007) can contribute to a rise in generator temperature and potential failure of the hydropower system. It can be observed that the factors affecting generator temperature are quite numerous and complex, making it unfeasible to analyze any single factor in isolation.

Deep learning, a subset of artificial intelligence (AI), has proven effective in tackling complex engineering challenges such as heat transfer, solving Reynolds boundary value problems, and addressing the shortcomings of traditional analytical methods (Almqvist, 2021; Goharoodi et al., 2019; He et al., 2021; Kamble et al., 2014; Vadyala et al., 2022; Zobeiry & Humfeld, 2021). Additionally, it has been employed to forecast the required outputs in various engineering applications (Bhattacharyya et al., 2021; Krishnayatra et al., 2020; Milan et al., 2021; Wang et al., 2020). Deep learning models are designed to manage intricate and incomplete datasets, increasing the reliability of solutions to engineering problems (Xu & Saleh, 2021). As a result, deep learning has minimized experimentation time while delivering more precise outcomes (Nasiri et al., 2019; Nilpueng et al., 2022).

Deep learning processes data in a way that resembles the cognitive functions of the human brain. It decomposes incoming data into smaller components and systematically analyzes them through multiple layers until output is generated. This output is then validated for accuracy against practical actual data. Deep learning models have diverse applications in heat transfer problems. Cai et al. (2021) employed deep learning methodologies to predict temperature distributions, providing practical solutions to heat transfer challenges characterized by ambiguous boundary conditions. Ghettni et al. (2020) investigated the efficacy of deep learning in assessing the performance of air-cooled condensers, determining that neural networks yield the most precise forecasts. Jadhav et

al. (2022) developed a neural network architecture designed to monitor the operational status of air preheaters (APH) in thermal power plants in real time, enabling accurate predictions of temperature profiles within the APH. Additionally, Chen et al. (2021) employed machine learning models to anticipate and enhance the performance of air-cooled condensers (ACC) within large-scale power facilities, illustrating that these models could deliver precise performance assessments of the ACC.

This study explores the use of deep learning to address the challenge of predicting stator winding temperatures in a 125 MW hydropower plant in Thailand. By applying artificial neural network models to operational and environmental data, the research demonstrates how data-driven approaches can provide accurate forecasts of thermal behavior and offer insights into the key factors influencing system performance. Such predictive capability is expected to support proactive maintenance and enhance the overall efficiency and reliability of hydropower operations.

2. Objectives

This research aims to develop and evaluate deep learning-based artificial neural network (ANN) models for predicting conductor winding temperatures in a 125 MW hydropower generator in Thailand. The study explores three ANN architectures Feedforward Neural Network (FNN), Multilayer Feedforward Neural Network (MFNN), and Long Short-Term Memory (LSTM) using data from 2018 to 2021, which includes generator output, reservoir head, flow rates, and thermal conditions. The model with the highest prediction accuracy will be selected, and the most influential input features will be identified to enhance system interpretability. This approach seeks to support proactive maintenance by enabling early detection of thermal stress trends, potentially reducing computational overhead and improving operational efficiency in hydropower applications.

3. Methodology

3.1 Data Preparation

This section analyzes the patterns and distribution of daily electricity generation data collected from 2018 to 2021, including:

1. Electricity generated GEN_ACT_POWER)
2. Water flow rate in the heat exchanger (WATER_FLOW)

3. Water Inlet temperature of the heat exchanger (WATER_INLET_TEMP)
4. Reservoir water level (HEAD)
5. Proxy for sediment accumulation (MONTH) classified step-by-step between 1-12 following the month number: 1 represents minimum and 12 represents maximum.
6. Temperature of the conductor windings in the generator (STATOR_WND_TEMP)
7. Air inlet temperature of the heat exchanger (GEN_AIR_IN)
8. Air outlet temperature of the heat exchanger (GEN_AIR_OUT)

The raw data of these variables were recorded at 5-minute intervals throughout the year as given in Figure 2. Prior to use, these data were normalized using the Min-Max Normalization technique (Kim et al., 2025) to scale all values within the range of 0 to 1. This step was performed to prevent variables with higher magnitudes from disproportionately influencing the learning process. Subsequently, the normalized data were transformed into a multidimensional structure compatible with PyTorch, enabling efficient

computation and preparation for inputs into the model.

In Figure 2, electricity generation, which already captured the combined effect of voltage (V), current (I), and power factor (PF) as $V \cdot I \cdot PF$ and conductor winding temperature increased continuously during the third and fourth quarters, then dropped in the first quarter of the following year. This pattern is repeated annually, corresponding with changes in reservoir water level. The inlet water temperature followed a trend opposite to the water level. Since direct measurements of sediment accumulation in the heat exchanger system were unavailable, an alternative proxy was used. The values were categorized incrementally from 1 to 12 following the month of the year (1 representing the lowest and 12 the highest). The inlet air temperature for the heat exchanger should remain below 60°C , while the outlet air temperature for the hydropower plant was designed to stay under 45°C . Notably, in the second and third quarters of 2019, there was a sharp increase in values due to high inlet water temperature and low water level, leading to elevated conductor winding temperatures and reduced electrical power generation.

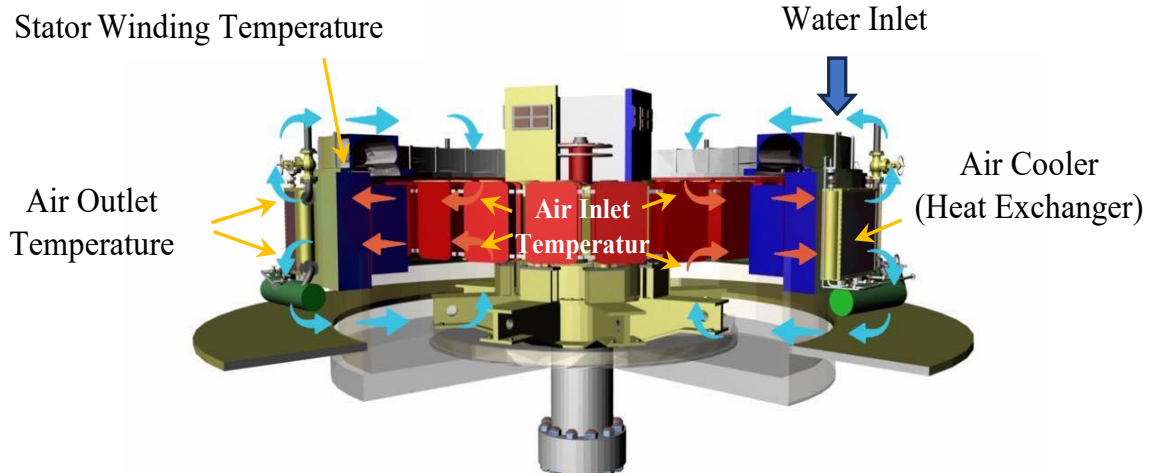


Figure 1 3D model of air cooler (heat exchanger) in stator windings of hydropower generator

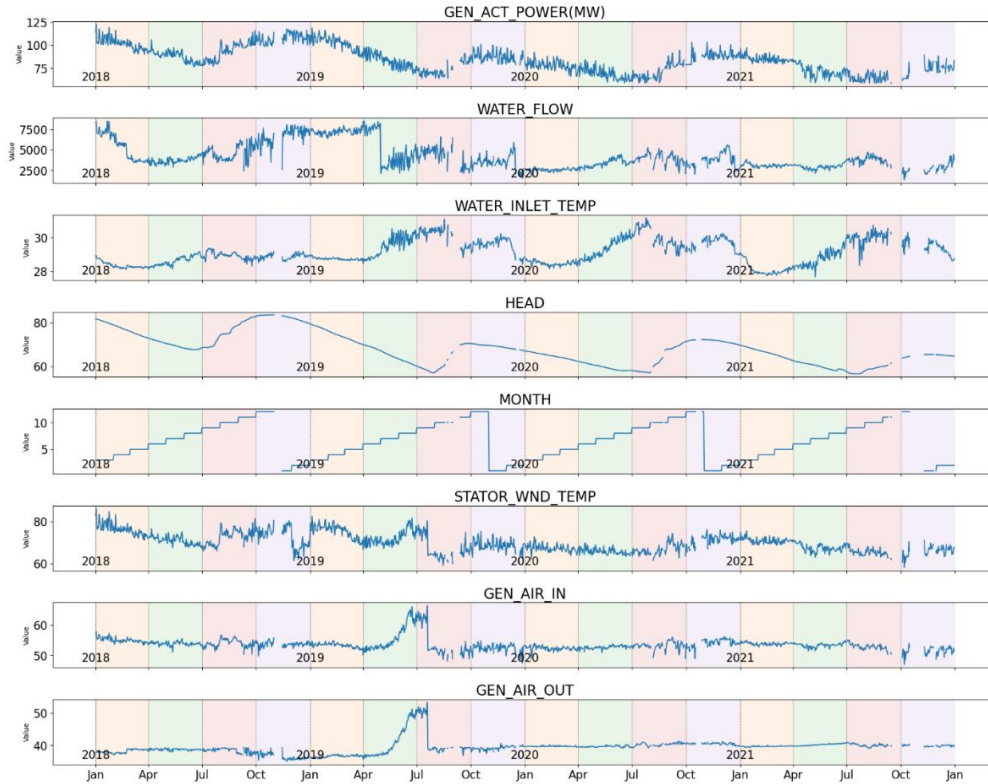


Figure 2 Data from 2018 to 2021

The graphs in Figure 2 used four distinct colors to represent quarterly data for each year. Missing data points in the graph were due to either irregular values or times when no electricity was generated. There are eight variables in total, with five that can be specified: GEN_ACT_POWER, WATER_FLOW, WATER_INLET_TEMP, HEAD, and MONTH. These act as input parameters. Consequently, the remaining variables, STATOR_WND_TEMP, GEN_AIR_IN, and GEN_AIR_OUT, will function as output parameters to be assessed. Table 1 shows the unit of all parameters.

The present study utilized data from 2018, 2019, and 2020. Data from 2018 was selected as the training dataset due to the high-water levels in reservoirs and the presumed absence of anomalies in the power generation system. The model performance was then validated using data from 2020. Additionally, the accuracy of the constructed model was further assessed using data from 2019, as it presented abnormalities with high inlet water temperatures and low water levels, leading to higher conductor winding temperatures and reduced electricity generation.

Table 1 Independent and Dependent Variables

Independent (inputs)	Unit
GEN_ACT_POWER	MW
WATER_FLOW	L/min
WATER_INLET_TEMP	°C
HEAD	m MSL
MONTH	-
Dependent (outputs)	Unit
STATOR_WND_TEMP	°C
GEN_AIR_IN	°C
GEN_AIR_OUT	°C

3.2 Neural Network Models

This section provides a comprehensive overview of the neural network models, constructed upon a foundational architecture consisting of multiple layers, namely the input, hidden, and output layers, as illustrated in Figure 3. The implementation of these models was carried out using Python in conjunction with the PyTorch framework (Paszke et al., 2019). The three models under discussion include the first model, identified as a Feedforward Neural Network (FNN); the second model, categorized as a Multilayer Feedforward Neural Network (MFNN);

and the third model, characterized as a Long Short-Term Memory (LSTM) Network.

3.2.1 Feedforward Neural Network Model (Ojha et al., 2017)

In this neural network, data flowed from the input layer to the output layer through hidden layers in a single direction without feedback. This could be represented mathematically as:

$$h(j) = f\left(\sum_{i=1}^m x_i w_{i,j}^{(1)} + b_j^{(1)}\right), \quad (1)$$

where $h(j)$ is the result of the j^{th} node in Hidden Layer, $f(\cdot)$ is the Rectified Linear Unit (ReLU) activation function (Ding et al., 2018), x_i is the input parameter in Input Layer ($i = 1, 2, \dots, m$ in this study),

$w_{i,j}^{(1)}$ is the synaptic weights matrix between the Input and Hidden Layers, $b_j^{(1)}$ is the bias of the j^{th} node in Hidden Layer.

$h(j)$ from the Hidden Layer is then transferred to the Output Layer for calculating the final output ($\hat{y}_k$) where k is the number of considered output ($k = 0, 1, 2$ in this study) as:

$$\hat{y}_k = \left(\sum_{j=1}^n h(j) w_{j,k}^{(2)} + b_k^{(2)}\right), \quad (2)$$

where $\hat{y}_k$ is the considered output, $w_{j,k}^{(2)}$ is the synaptic weights matrix between the Hidden Layers and Output Layer, $b_k^{(2)}$ is the bias of the k^{th} node in Output Layer, n is the number of hidden layer nodes.

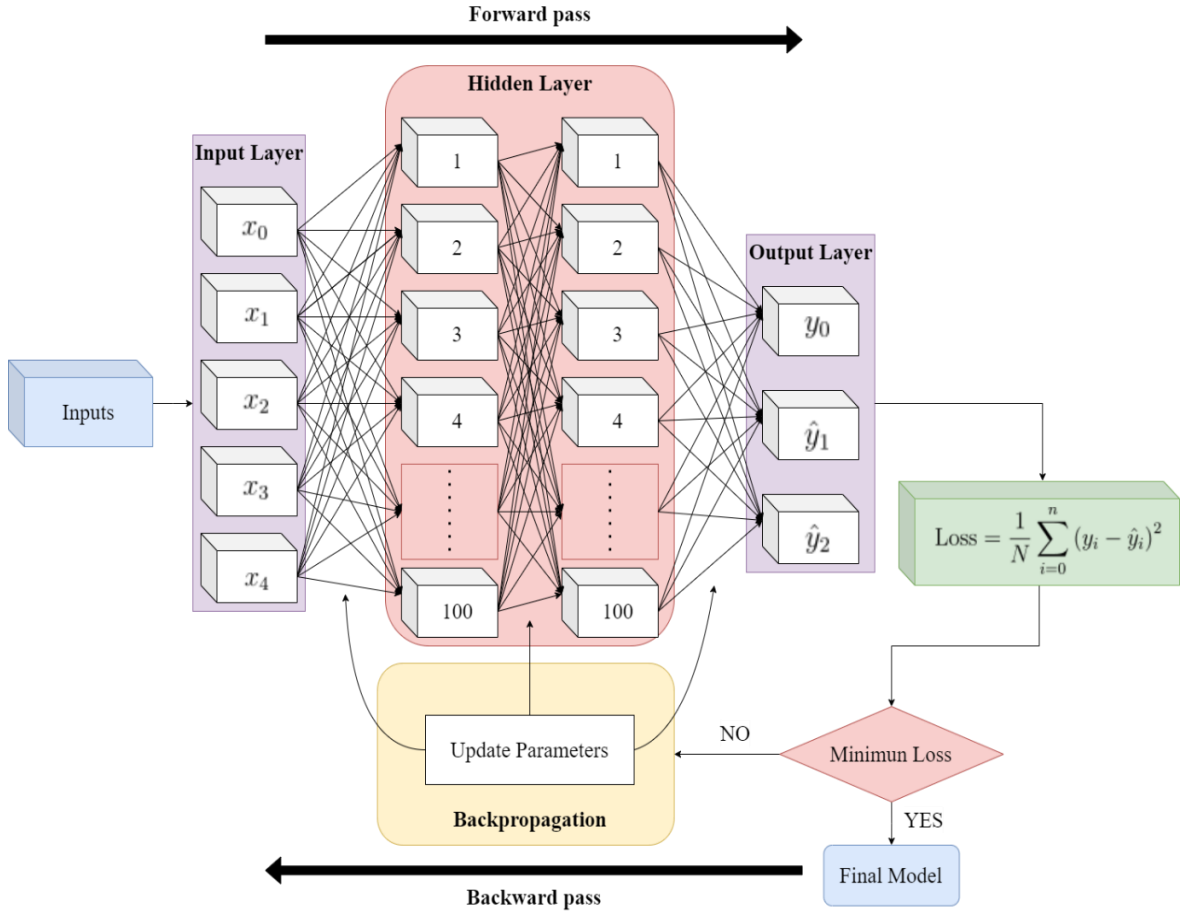


Figure 3 The structure of the neural network model

3.2.2 Multilayer Feedforward Neural Network Model (Le et al., 2023)

This framework is built on the standard feedforward neural network architecture, with modifications that involve adding extra hidden layers between the input and output layers. These changes significantly boost the model's ability to effectively capture complex features. In this study, two hidden layers were used, which can be expressed mathematically as:

$$h_1(j) = f\left(\sum_{i=1}^m x_i w_{i,j}^{(1)} + b_j^{(1)}\right), \quad (3)$$

$$h_2(k) = f\left(\sum_{j=1}^n h_1(j) w_{j,k}^{(2)} + b_k^{(2)}\right), \quad (4)$$

$$\hat{y}_l = \left(\sum_{k=1}^p h_2(k) w_{k,l}^{(3)} + b_l^{(3)}\right), \quad (5)$$

where $h_1(j)$ is the result of the j^{th} node in Hidden Layer 1 and $h_2(k)$ is the result of the k^{th} node in Hidden Layer 2, $b_k^{(2)}$ is the bias of the k^{th} node in Hidden Layer 2, $\hat{y}_l$ is the considered output, $w_{j,k}^{(2)}$ is the synaptic weight matrix between the Hidden Layers 1 and 2, $w_{k,l}^{(3)}$ is the synaptic weight matrix between the Hidden Layer 2 and Output Layer, $b_l^{(3)}$ is the bias

of the l^{th} node in Output Layer, m and p are the number of hidden layer nodes in Hidden Layer 1 and 2, respectively.

3.2.3 Long Short-Term Memory Model

This model is built on the foundational structure of the Feedforward Neural Network (FNN) to enhance predictive performance. However, FNN algorithms face limitations in handling long data sequences with complex, continuous relationships, as they lack the ability to retain previous information for future computations (Zhang et al., 2018). To overcome this, Long Short-Term Memory (LSTM) networks are introduced, incorporating memory cells and gate mechanisms. These features enable LSTM networks to effectively retain essential information and discard irrelevant data in sequences. The flow of information in LSTM networks is controlled by three internal gates: the forget gate, the input gate, and the output gate, as shown in Figure 4. The forget gate determines which information in the cell state should be discarded, the input gate decides what new information to add, and the output gate regulates the output of the hidden layer. This gating mechanism allows LSTM networks to efficiently manage long-term dependencies by selectively updating and retaining information throughout the learning process.

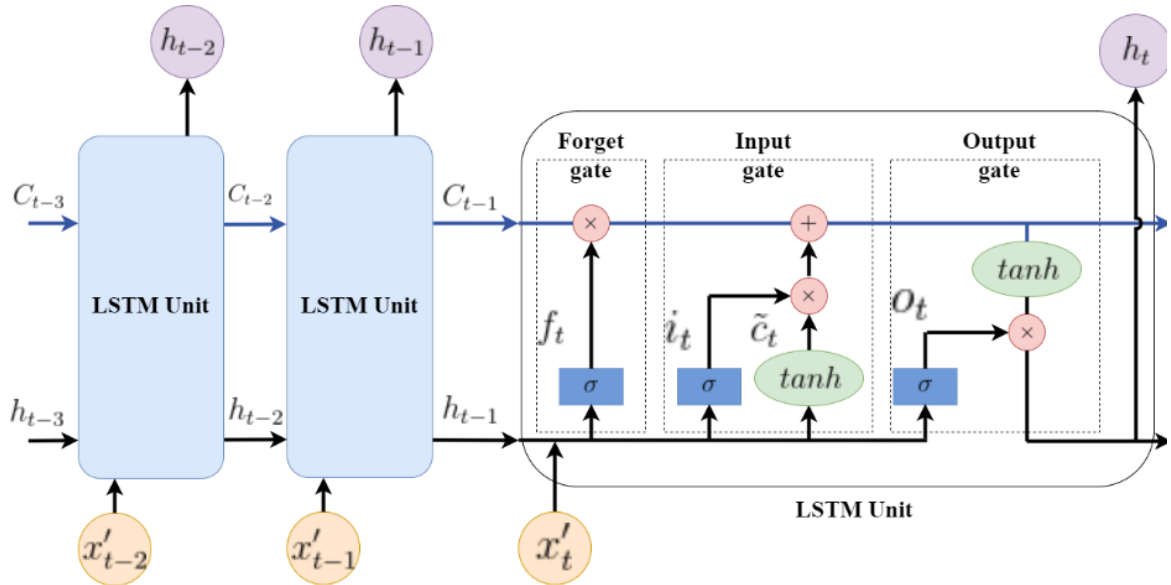


Figure 4 Schematic of the internal structure of an LSTM unit (Al-Selwi et al., 2024)

In this study, the calculation progresses from the Input Layer to the LSTM Layer (Hidden Layer), a linear transformation matches the input dimension from 5 to the size of the Hidden Layer with 200 nodes, and it is performed as:

$$x'_t = w_{in} x_t + b_{in}, \quad (6)$$

where x_t is the input data at timestep t which consists of multiple timesteps:

$x_t = [x_{t,0}, x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4}]$, w_{in} is the weight matrix for transforming the input data to the hidden layer, b_{in} is the bias vector for the transformation, x'_t is the transformed input data.

In the hidden layer of LSTM model, x'_t is derived from linear transformations to perform computations through the forget gate, input gate, and output gate. These gates work together to update both the hidden state (h_t), and the cell state (c_t) as follows:

- **Forget Gate:** The forget gate determines how much information from the previous cell state (c_{t-1}) should be retained or forgotten which can be evaluated by

$$f_t = \sigma(w_f \cdot [h_{t-1}, x'_t] + b_f), \quad (7)$$

where f_t is the forget gate value at timestep, w_f is the forget gate weights, b_f is the forget gate biases, h_{t-1} is the hidden state from previous timestep $t-1$, σ represents the sigmoid activation function, which ensures that the forget gate values range between 0 and 1. When the f_t value is close to 1, the cell state preserves the corresponding information. Conversely, when the f_t value is close to 0, the cell state forgets that information.

- **Input Gate:** The input gate determines which new information should be added to the cell state, which computes the input gate activation by

$$i_t = \sigma(w_i \cdot [h_{t-1}, x'_t] + b_i), \quad (8)$$

and that of the candidate cell state by

$$\tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x'_t] + b_c), \quad (9)$$

then the cell state was updated by combining the retained information ($f_t \cdot c_{t-1}$) and the new candidate cell state ($i_t \cdot \tilde{c}_t$) as:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \quad (10)$$

where i_t is the input gate value at timestep t , w_i is the input gate weights, b_i is the input gate biases, $\tilde{c}_t$ is the new cell state data at timestep t , w_c is the candidate cell state weights, b_c is the candidate cell state biases, c_t is the updated cell state at timestep t , c_{t-1} is the previous timestep cell state $t-1$. The hyperbolic tangent ($\tanh$) activation function compresses values between -1 and 1, helping in balancing the updates to the cell state. If the i_t value is close to 1, it strongly contributes new information to the cell state, and if the i_t value is near 0, the addition of new information is minimized.

- **Output Gate:** The output gate determines the next hidden state (h_t) based on the updated cell state and the input, which calculates the output gate activation by

$$o_t = \sigma(w_o \cdot [h_{t-1}, x'_t] + b_o), \quad (11)$$

and computes the hidden state as:

$$h_t = o_t \cdot \tanh(c_t), \quad (12)$$

where o_t is the output gate value which is controlled by sigmoid function to pass from the cell state to hidden state, h_t is the updated hidden state at timestep t which is used as the input for the next timestep or passed to the output layer for prediction. c_t is the updated cell state at the current timestep which is modulated with $\tanh$ to limit its range before multiplying with the output gate values.

The gates collaboratively regulate the flow of information, ensuring long-term dependencies are effectively captured and short-term irrelevant information is forgotten.

Finally, the final output of the LSTM unit ($\hat{y}$) is usually a function of the hidden state (h_t) from the LSTM layer and additional parameters (w_{out}) as:

$$\hat{y} = h_t w_{out} + b_{out}, \quad (13)$$

where $\hat{y}_t$ is the model output at timestep t . w_{out} is the output layer weights, and b_{out} is the output layer biases.

Once the predicted output is obtained from each model architecture, the training process aims to minimize the discrepancy between predicted and actual values through iterative updates of the model's weights and biases. This optimization is performed using backpropagation and a gradient-based update method.

During each epoch, a full pass through the training dataset, the model processes every example once and computes the loss by comparing the predicted values ($\hat{y}_i$) with the true values (y_i). The loss function used is the Mean Squared Error (MSE), defined as:

$$Loss = MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (14)$$

where N represents the number of samples used in each training iteration.

If the loss remains high, the model updates its parameters using the gradient of the loss with respect to each parameter. The conceptual update rules for the weights (w_{t+1}) and biases (b_{t+1}) are

$$w_{t+1} = w_t - \alpha \cdot \frac{\partial Loss}{\partial w_t}, \quad (15)$$

$$b_{t+1} = b_t - \alpha \cdot \frac{\partial Loss}{\partial b_t}, \quad (16)$$

where α is the learning rate.

The training parameters for the neural network models are configured using PyTorch Lightning within a supervised learning framework. Prior to training, the learning rate is selected using the lr_finder utility to identify a suitable range based on loss trends.

The models were trained using the Adam optimizer with an initial learning rate of 0.001. The training process was set to run for up to 1000 epochs, with early stopping applied using a patience of 10 epochs to halt training when the validation loss no longer showed improvement. In addition, a ReduceLROnPlateau scheduler was employed to dynamically adjust the learning rate.

To preserve the best-performing model, ModelCheckpoint (PyTorch Lightning team, n.d.) was used to save the model with the lowest validation loss. Training initially began with 50 nodes in the hidden layer, which were then gradually increased until prediction accuracy stabilized typically around 100 nodes. Parameters not explicitly mentioned were kept at their PyTorch Lightning default values.

4. Results

4.1 Prediction Results

4.1.1 Results of the FNN Model

Figure 5 provides a comparison between empirical data and predictions generated by the FNN model (Model 1) utilizing 100 and 200 nodes in the hidden layer for the variables STATOR_WND_TEMP, GEN_AIR_IN, and GEN_AIR_OUT. The findings indicated that the model with a greater number of nodes demonstrated an enhanced ability to capture complex patterns. Specifically, the analysis revealed that the 200-node model achieved superior predictive performance compared to the 100-node model. However, the improvement was marginal, suggesting that the 200-node model remained sufficiently accurate for the given task. The predictive performance was quantified using the Mean Squared Error (MSE), with the 100 and 200-node models (FNN) achieving average MSE value of 0.02868 and 0.02580, respectively, highlighting their satisfactory accuracy.

4.1.2 Results of the MFNN Model

The MFNN model (Model 2) was constructed as an extension of the FNN model, incorporating two hidden layers with 100 nodes in each layer. This architectural enhancement proved effective across a range of variables, including those exhibiting lower volatility, such as STATOR_WND_TEMP and GEN_AIR_IN_TEMP, as well as the more variable GEN_AIR_OUT_TEMP. As shown in Figure 6, Model 2 demonstrated a markedly improved ability to capture and track data trends across all variables compared to Model 1. Furthermore, the MFNN model achieved a lower average Mean Squared Error (MSE) of 0.02278, signifying a notable improvement in predictive accuracy. The increased complexity of the model's architecture enabled enhanced data processing capabilities, particularly under conditions of higher variability.

4.1.3 Results of the LSTM Model

The LSTM model (Model 3), which also featured 200 nodes in its hidden layer, was utilized to generate predictions for STATOR_WND_TEMP, GEN_AIR_IN, and GEN_AIR_OUT. Model 3 demonstrated exceptional proficiency in producing predictions closely aligned with the observed data. This performance was attributed to its recurrent neural network (RNN) architecture, which enabled the model to effectively learn and capture sequential relationships within the dataset, distinguishing it from

traditional feedforward neural network (FNN) architectures, as illustrated in Figure 7. The LSTM model achieved the lowest average Mean Squared Error (MSE) of 0.00373, establishing itself as the most accurate predictive model among those

evaluated. Its architecture showcased significant potential for handling time-series data, effectively identifying patterns and trends with a level of precision superior to that of the other models.

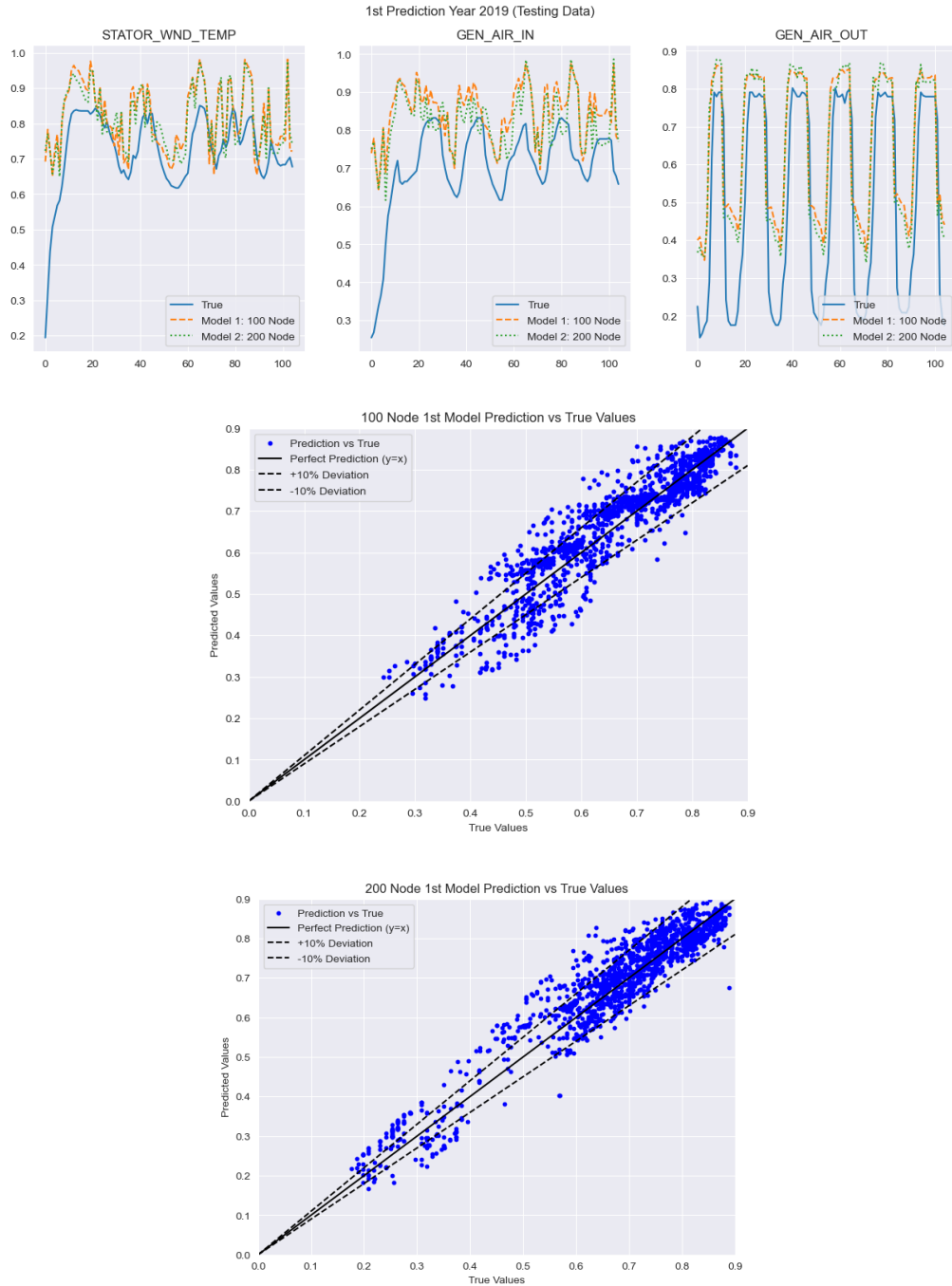


Figure 5 Prediction results of the first model (FNN) with 100 and 200 nodes in the year 2019

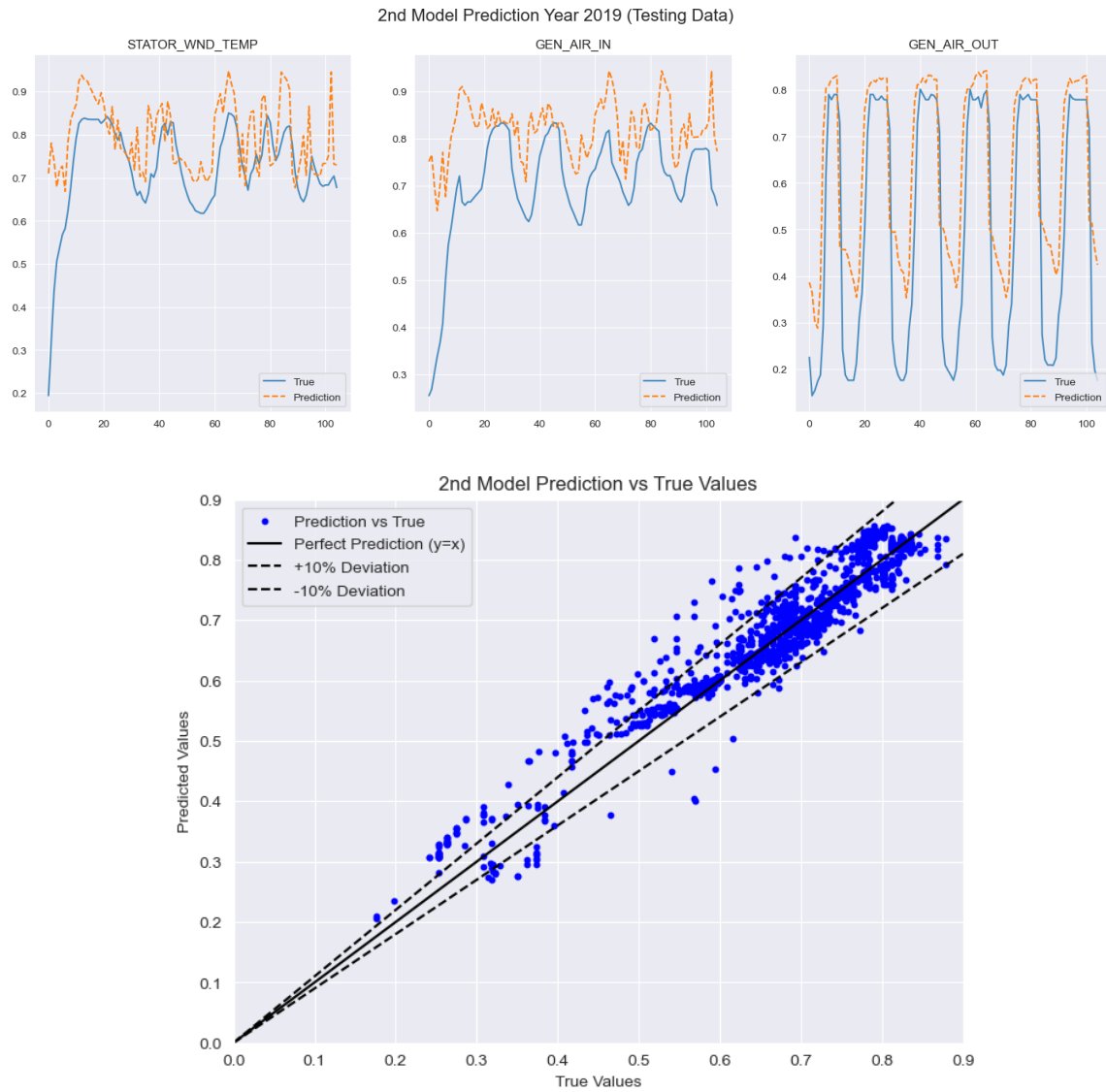


Figure 6 Prediction results of the MFNN model (Model 2) in the year 2019

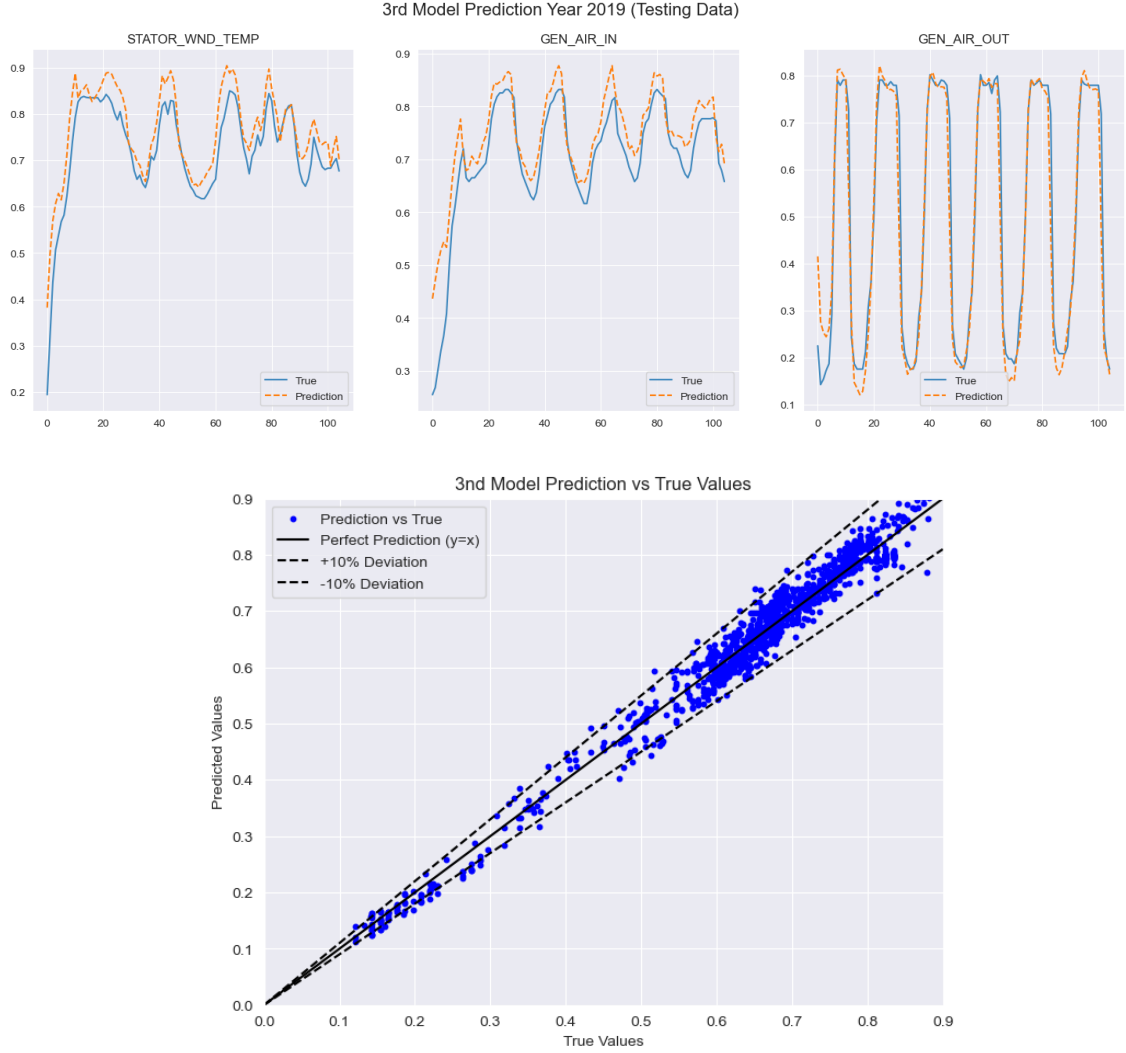


Figure 7 Prediction results in the LSTM model (Model 3) in the year 2019

The LSTM model was further evaluated through forward testing using data from the year 2021, which represented unseen data. The results demonstrated high predictive accuracy, with a Mean Squared Error (MSE) of less than 0.0053, highlighting the model reliability and robustness in forecasting future thermal behavior.

4.2 Effects of Parameter Inputs on Outputs

Identifying the input parameters that most significantly influence model outputs is a complex task when utilizing ANN models. The Shapley Additive exPlanations (SHAP) method is a commonly adopted approach to address this challenge (Rodríguez-Pérez & Bajorath, 2020). By calculating absolute SHAP

values, the contribution of individual features to model predictions can be systematically evaluated. In this study, the analysis utilizes predicted data obtained from the LSTM model outlined in the preceding section. The SHAP values can be computed as follows (Chen et al., 2022):

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \quad (17)$$

where ϕ_i is the SHAP value of input parameter i , representing its average contribution to the model prediction. F denotes the set of all input parameters in the model, and S is a subset of F that excludes the input parameter i , with $|S|$ indicating its cardinality. The function $f(S)$ corresponds to the model prediction

based on the subset S , while $f(S \cup \{i\})$ represents the model prediction when i is included in the subset S . Additionally, $\frac{|S|!(|F|-|S|-1)!}{|F|!}$ is a weighting factor applied to ensure a fair distribution of contributions across different subsets.

Figure 8 displays the SHAP value plots, highlighting the mean SHAP values assigned to each input parameter to reflect their relative importance in the model's predictive analysis. Additionally, a color gradient is used to visually convey the influence of these parameters on various output variables: GEN_AIR_OUT (illustrated in blue), GEN_AIR_IN (depicted in pink), and STATOR_WND_TEMP (represented in green). Figure 8a, 8b, 8c, and 8d illustrate the impacts of input parameters on the prediction values for the years 2018, 2019, 2020, and 2021, respectively.

The Figures illustrate that the STATOR_WND_TEMP and GEN_AIR_OUT are predominantly influenced by the GEN_ACT_POWER, WATER_FLOW, and WATER_INLET_TEMP. In particular, the GEN_AIR_OUT temperature is primarily impacted by the WATER_FLOW and WATER_INLET_TEMP. However, in 2019, the WATER_INLET_TEMP had the most significant effect on the

STATOR_WND_TEMP. This was due to a mismatch between the water flow rate adjustment and the power generation capacity, coupled with an increase in WATER_INLET_TEMP during the seasonal period (see Figure 2, April–October 2019). Furthermore, during this period, the GEN_AIR_IN temperature rose, leading to a corresponding increase in GEN_AIR_OUT. The combined effect of these temperature increases likely contributed to a shutdown of the hydroelectric power plant. In contrast, in 2020 and 2021, the synchronization of WATER_FLOW adjustments with GEN_ACT_POWER and WATER_INLET_TEMP could avoid similar issues.

As mentioned above, it is evident that the model derived from the Artificial Neural Network (ANN) can accurately predict the GEN_AIR_IN, GEN_AIR_OUT, and Stator winding temperatures of the hydropower plant. Additionally, the SHAP value data can be utilized to analyze the impact of input variables on these three outputs, providing valuable insights for improving the generator's cooling system efficiency. Furthermore, this analysis can support the planning of preventive maintenance strategies, ensuring the optimal operation and performance of the power generation system.

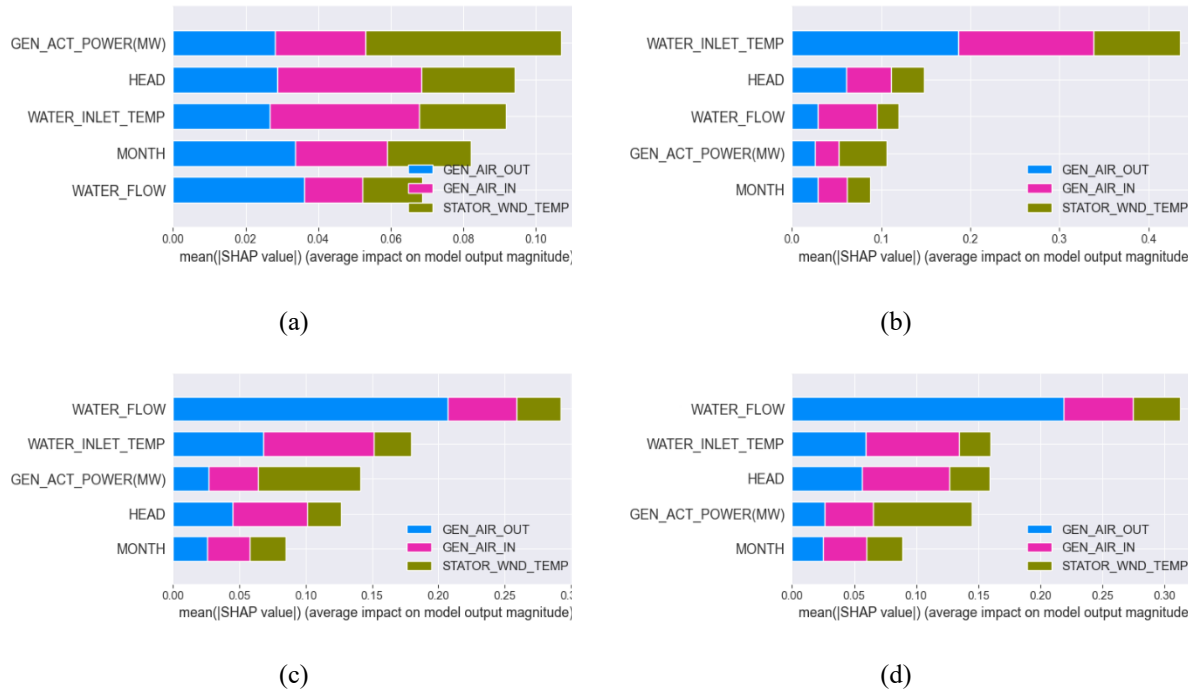


Figure 8 The impacts of input parameters on the prediction values: (a) Data in year 2018, (b) Data in year 2019 (c) Data in year 2020, (d) Data in year 2021

5. Conclusion

Deep learning can be applied to analyze stator winding temperature variations and cooling air temperature in the heat exchanger of a 125 MW hydropower plant in Thailand. Given the numerous interrelated variables and their complex relationships, conventional analytical methods are insufficient for effectively addressing this challenge. The implementation of deep learning provides an innovative approach, enabling operators to gain deeper insights into the factors influencing stator winding temperature and identify potential operational issues. The key findings of this study are summarized as follows:

- In hydropower generation, copper and core losses at the stator windings convert to heat, making temperature control essential to prevent shutdowns and damage. Stator temperature can rise due to factors like load surges, cooling water fouling, water flow fluctuations, temperature changes, weather conditions, and reservoir levels. These complex interactions make it impractical to analyze any single factor in isolation. Artificial Neural Network models can predict temperature trends and identify critical factors affecting system behavior. This helps administrators and maintenance teams anticipate issues and optimize operations before problems arise.

- The FNN model with 100 and 200 nodes in the hidden layer was trained on 2018 data and validated with 2020 data to learn complex patterns. Training, conducted using PyTorch Lightning within a supervised framework, was tested from 10 to 1000 epochs until the predictions stabilized and closely aligned with the actual data. A learning rate finder dynamically adjusted the rate, and early stopping prevented overfitting. The analysis revealed that the 200-node model slightly outperformed the 100-node model. Both models achieved satisfactory accuracy, with MSE values of 0.02580 for the 200-node model and 0.02868 for the 100-node model.

- The MFNN and LSTM models followed a similar training process as the FNN model, using 200 nodes in the hidden layer. Designed specifically for time-series data, the LSTM model exhibited the highest predictive accuracy, achieving the lowest Mean Squared Error of 0.00373, compared to 0.02868 for the FNN model and 0.02278 for the MFNN model.

- Shapley Additive Explanations is an effective method for assessing the variables influencing stator winding temperature in a hydropower generator. The study indicates that under normal operating conditions, the most significant factor affecting stator

winding temperature is the generated active power, followed by water flow rate and inlet water temperature, respectively. However, under abnormal operating conditions, the influence of these variables may shift. For instance, in 2019, the inlet water temperature was identified as the primary factor impacting stator winding temperature.

5.1 Further Studies

This study can be expanded to practical applications under various conditions within hydropower plants, including:

- Modeling operations across various scenarios: The model can simulate plant performance under different conditions, including changes in reservoir water levels, water temperature, and power generation. This allows operators to proactively assess and mitigate potential impacts.

- Improving efficiency and optimizing performance: The model can be applied to predict and enhance energy generation efficiency by estimating power output based on water flow adjustments and by optimizing heat exchanger cooling performance. Furthermore, it can aid in implementing proactive maintenance strategies.

6. Acknowledgments

The authors express their gratitude to the Department of Mechanical Engineering, Faculty of Engineering, Chiang Mai University, and the Electricity Generating Authority of Thailand (EGAT) for their invaluable financial support of this research. Special thanks to Professor Subpasit Chongsawat for his assistance with content composition and guidance in developing the model's operational processes. Additionally, appreciation is conveyed to Mr. Prakobkit Sabaywong for his contributions to the design of the 3D images of the hydropower plant.

7. CRediT Statement

Chinachote Deevijit: Software; validation; formal analysis; investigation; resources; data curation; writing, original draft.

Tanongkiat Kiatsiriroat: Conceptualization; methodology; formal analysis; writing, review & editing; visualization; supervision.

Thoranis Deethayat: Validation; formal analysis; investigation; resources; data curation; supervision.

Attakorn Asanakham: Conceptualization; methodology; validation; formal analysis; investigation; data curation; writing, review & editing; supervision.

8. Reference

- Aarons, K., Vine, D., & Center for Climate and Energy Solutions. (2015). *Canadian hydropower and the Clean Power Plan*. Retrieved from <https://www.c2es.org/document/canadian-hydropower-and-the-clean-power-plan/>
- Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., & Ragab, M. G. (2024). RNN-LSTM: From applications to modeling techniques and beyond Systematic review. *Journal of King Saud University-Computer and Information Sciences*, 36(5), Article 102068. <https://doi.org/10.1016/j.jksuci.2024.102068>
- Almqvist, A. (2021). Fundamentals of physics-informed neural networks applied to solve the Reynolds boundary value problem. *Lubricants*, 9(8), Article 82. <https://doi.org/10.3390/lubricants9080082>
- Asvapoositkul, W., & Kuansathan, M. (2016). The effectiveness-NTU approach for evaluation of an air-cooled heat exchanger. *Heat Transfer Engineering*, 37(2), 140–149. <https://doi.org/10.1080/01457632.2015.1044393>
- Bhattacharyya, S., Sarkar, D., Roy, R., Chakraborty, S., Goel, V., & Almatrafi, E. (2021). Application of new artificial neural network to predict heat transfer and thermal performance of a solar air-heater tube. *Sustainability (Switzerland)*, 13(13), Article 7477. <https://doi.org/10.3390/su13137477>
- Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6), Article 060801. <https://doi.org/10.1115/1.4050542>
- Chen, H., Lundberg, S. M., & Lee, S. I. (2022). Explaining a series of models by propagating Shapley values. *Nature Communications*, 13(1), Article 31384. <https://doi.org/10.1038/s41467-022-31384-3>
- Chen, H., Peng, W., Nie, C., Xu, G., & Lei, J. (2021). Performance prediction and optimization of the air-cooled condenser in a large-scale power plant using machine learning. *Energy Technology*, 9(11), Article 2100045. <https://doi.org/10.1002/ente.202100045>
- Cui, Z., Li, N., & Wu, L. (2021). Multi-objective optimization of the generator air cooler based on genetic algorithm. *Journal of Physics: Conference Series*, 2030, Article 012079. <https://doi.org/10.1088/1742-6596/2030/1/012079>
- Desingu, K., Chelliah, T. R., & Khare, D. (2018). Investigation on the performance of a 277.8 MVA synchronous air-cooled hydrogenerator through loss models [Conference presentation]. *2018 International Conference on Power, Instrumentation, Control and Computing (PICC)*, Thrissur, India. <https://doi.org/10.1109/PICC.2018.8384790>
- Ding, B., Qian, H., & Zhou, J. (2018). Activation functions and their characteristics in deep neural networks [Conference presentation]. *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, Shenyang, China. <https://doi.org/10.1109/CCDC.2018.8407425>
- Doost, A. K., & Majlessi, R. (2015). Heat transfer analysis in cooling system of hydropower's generator. *Open Journal of Applied Sciences*, 5(3), 98–107. <https://doi.org/10.4236/OJAPPS.2015.53010>
- Electricity Generating Authority of Thailand. (n.d.). *Renewable energy*. Retrieved from <https://www.egat.co.th/home/renewables/>
- Ghettini, S., Sorce, A., & Sacile, R. (2020). Data-driven air-cooled condenser performance assessment: Model and input variable selection comparison. *E3S Web of Conferences*, 197, Article 10003. <https://doi.org/10.1051/e3sconf/202019710003>
- Goharoodi, S. K., Nguyen Phuc, P., Dupre, L., & Crevecoeur, G. (2019). Data-driven discovery of the heat equation in an induction machine via sparse regression [Conference presentation]. *Proceedings of the IEEE International Conference on Industrial Technology*, Melbourne, VIC, Australia. <https://doi.org/10.1109/ICIT.2019.8754983>
- He, Z., Ni, F., Wang, W., & Zhang, J. (2021). A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials. *Materials Today Communications*, 28, Article 102719. <https://doi.org/10.1016/j.mtcomm.2021.102719>
- Jadhav, V., Deodhar, A., Gupta, A., & Runkana, V. (2022). Physics-informed neural network for health monitoring of an air preheater. *PHM Society European Conference*, 7(1), 219–230. <https://doi.org/10.36001/phme.2022.v7i1.3343>

- Kamble, L. V., Pangavhane, D. R., & Singh, T. P. (2014). Heat transfer studies using artificial neural network – A review. *International Energy Journal*, 14(1), 25–42. <http://www.ericjournal.ait.ac.th/index.php/eric/article/view/1066>
- Khun, N. S., Deethayat, T., Kiatsiriroat, T., & Asanakham, A. (2025). Strategy for energy savings in a commercial building air-conditioning system with chilled water storage: A case study in a retail mall in Thailand. *Journal of Current Science and Technology*, 15(1), Article 89. <https://doi.org/10.59796/jcst.V15N1.2025.89>
- Kim, Y.-S., Kim, M. K., Fu, N., Liu, J., Wang, J., & Srebric, J. (2025). Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models. *Sustainable Cities and Society*, 118, Article 105570. <https://doi.org/10.1016/j.scs.2024.105570>
- Krishnayatra, G., Tokas, S., & Kumar, R. (2020). Numerical heat transfer analysis & predicting thermal performance of fins for a novel heat exchanger using machine learning. *Case Studies in Thermal Engineering*, 21, Article 100706. <https://doi.org/10.1016/j.csite.2020.100706>
- Le, X. K., Wang, N., & Jiang, X. (2023). Nuclear mass predictions with multi-hidden-layer feedforward neural network. *Nuclear Physics A*, 1038, Article 122707. <https://doi.org/10.1016/j.nuclphysa.2023.122707>
- Milan, P. J., Hickey, J. P., Wang, X., & Yang, V. (2021). Deep-learning accelerated calculation of real-fluid properties in numerical simulation of complex flowfields. *Journal of Computational Physics*, 444, Article 110567. <https://doi.org/10.1016/j.jcp.2021.110567>
- Nasiri, A., Taheri-Garavand, A., Omid, M., & Carlomagno, G. M. (2019). Intelligent fault diagnosis of cooling radiator based on deep learning analysis of infrared thermal images. *Applied Thermal Engineering*, 163, Article 114410. <https://doi.org/10.1016/j.applthermaleng.2019.114410>
- Navarro, H. A., & Cabezas-Gómez, L. C. (2007). Effectiveness-ntu computation with a mathematical model for cross-flow heat exchangers. *Brazilian Journal of Chemical Engineering*, 24(4), 509–521. <https://doi.org/10.1590/S0104-66322007000400005>
- Nilpueng, K., Kaseethong, P., Mesgarpour, M., Shadloo, M. S., & Wongwises, S. (2022). A novel temperature prediction method without using energy equation based on physics-informed neural network (PINN): A case study on plate-circular/square pin-fin heat sinks. *Engineering Analysis with Boundary Elements*, 145, 404–417. <https://doi.org/10.1016/j.enganabound.2022.09.032>
- Nogueira, É., & Nogueira, É. (2022). Efficiency and effectiveness method versus ϵ -NTU method with application in finned flat tube compact heat exchanger with water-ethylene glycol as nanofluid base of iron oxide nanoparticles. *Journal of Materials Science and Chemical Engineering*, 10(2), 1–17. <https://doi.org/10.4236/MSCE.2022.102001>
- Ojha, V. K., Abraham, A., & Snašel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60, 97–116. <https://doi.org/10.1016/j.engappai.2017.01.013>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32. <https://doi.org/10.48550/arXiv.1912.01703>
- PyTorch Lightning team. (n.d.). *Introducing multiple ModelCheckpoint callbacks*. PyTorch Lightning Developer Blog. Retrieved from <https://devblog.pytorchlightning.ai/introducing-multiple-modelcheckpoint-callbacks-e4bc13f9c185>
- Rodríguez-Pérez, R., & Bajorath, J. (2020). Interpretation of machine learning models using shapley values: Application to compound potency and multi-target activity predictions. *Journal of Computer-aided Molecular Design*, 34(10), 1013–1026. <https://doi.org/10.1007/s10822-020-00314-0>
- Vadyala, S. R., Betgeri, S. N., & Betgeri, N. P. (2022). Physics-informed neural network method for solving one-dimensional advection equation using PyTorch. *Array*, 13, Article 100110. <https://doi.org/10.1016/j.array.2021.100110>

- Wang, W., Deng, S., Zhao, D., Zhao, L., Lin, S., & Chen, M. (2020). Application of machine learning into organic Rankine cycle for prediction and optimization of thermal and exergy efficiency. *Energy Conversion and Management*, 210, Article 112700. <https://doi.org/10.1016/j.enconman.2020.112700>
- Xu, Z., & Saleh, J. H. (2021). Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliability Engineering and System Safety*, 211, Article 107530. <https://doi.org/10.1016/j.res.2021.107530>
- Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a long short-term memory (LSTM)-based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, 561, 918–929. <https://doi.org/10.1016/j.jhydrol.2018.04.065>
- Zobeiry, N., & Humfeld, K. D. (2021). A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence*, 101, Article 104232. <https://doi.org/10.1016/j.engappai.2021.104232>

9. Nomenclature

$b_j^{(1)}$	The bias of the j^{th} node in Hidden Layer
$b_l^{(3)}$	The bias of the l^{th} node in Output Layer
b_{in}	The bias vector for the transformation
b_f	The forget gate biases
b_i	The input gate biases
b_c	The candidate cell state biases
b_o	The output gate biases
b_{out}	The output layer biases
b_t	The biases at the t^{th} iteration
b_{t+1}	The biases term at iteration $t+1$, updated via gradient descent
c_t	The updated cell state at timestep t
c_{t-1}	The previous timestep cell state $t-1$
$\tilde{c}_t$	The new cell state data at timestep t
F	The set of all input parameters in the model
f_t	The forget gate value at timestep
$h(j)$	The result of the j^{th} node in Hidden Layer
$h_1(j)$	The result of the j^{th} node in Hidden Layer 1
$h_2(k)$	The result of the k^{th} node in Hidden Layer 2
h_t	The updated hidden state at timestep t
h_{t-1}	The hidden state from previous timestep $t-1$
i_t	The input gate value at timestep t
$Loss$	The loss value that measures the discrepancy between the actual value and the predicted value
MSE	The Mean Squared Error is used to evaluate the loss
o_t	The output gate value at timestep t
S	The subset of F that excludes the input parameter i
$\tanh$	The hyperbolic tangent activation function
$w_{i,j}^{(1)}$	The synaptic weights matrix between The Input and Hidden Layers
$w_{j,k}^{(2)}$	The synaptic weight matrix between The Hidden Layers 1 and 2 (Also used to denote the weights between Hidden Layer 2 and the Output Layer in another equation.)

$w_{k,l}^{(3)}$	The synaptic weight matrix between The Hidden Layer 2 and Output Layer
w_{in}	The weight matrix for transforming the input data to the hidden layer
w_f	The forget gate weights
w_i	The input gate weights
w_c	The candidate cell state weights
w_o	The output gate weights
w_{out}	The output layer weights
w_t	The weights at the t^{th} iteration
w_{t+1}	The weights term at iteration $t+1$, updated via gradient descent
x_i	The input parameter in Input Layer ($i = 0, 1, 2, \dots m$ in this study)
x_t	The input data at timestep, which consists of multiple timesteps
x'_t	The transformed input data
$\hat{y}_k$	FNN output
$\hat{y}_l$	MFNN output
$\hat{y}_t$	LSTM output
y_i	The predicted values at position i
$\hat{y}_i$	The actual values at position i
ϕ_i	The SHAP value of an input parameter i
σ	The sigmoid activation function
α	The learning rate