Comparative Analysis of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) Models for Gas Turbine Performance Prediction

Abdul Rohman Rusdan Arif [1], Noor Akhmad Setiawan [2*], Joko Waluyo[3]

[1]Master of System Engineering, Faculty of Engineering, Universitas Gadjah Mada, Indonesia
[2]Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Indonesia
[3]Department of Mechanical and Industrial Engineering, Faculty of Engineering, Universitas Gadjah Mada, Indonesia

[*]Email address of Corresponding author: noorwewe@ugm.ac.id

ABSTRACT

Gas turbines are essential for offshore operations in the oil and gas industry due to their lightweight structure and high efficiency. Conventional maintenance relying on parameter monitoring and engine washing often causes unplanned downtime and suboptimal recovery. This study develops gas turbine performance prediction models using Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) deep learning approaches. Historical operational data were preprocessed through cleaning, normalization, and feature selection using Random Forest and LASSO. The LSTM model achieved an RMSE of 3.96 and an R^2 of 0.9991, while the GRU model achieved an RMSE of 4.58 and an R^2 of 0.9988. Comparative analysis showed that LSTM slightly outperformed GRU in accuracy, although GRU converged faster. These findings demonstrate the potential of integrating deep learning methods into predictive maintenance frameworks to enhance gas turbine reliability and efficiency.

Keywords: Gas Turbine, Performance Prediction, LSTM, GRU, Deep Learning

1. Introduction

Gas turbines play a pivotal role in offshore oil and gas operations due to their light weight and high operational efficiency. However, traditional maintenance approaches—relying on real-time parameter monitoring and scheduled engine washing—often result in unplanned downtime and suboptimal recovery. Moreover, conventional thermodynamic models struggle to capture the nonlinear behavior of gas turbines under varying conditions. Several studies have introduced more advanced predictive techniques. High Dimensional Model Representation (HDMR) and Artificial Neural Networks (ANN) for performance prediction [1]. Bayesian hierarchical model for Remaining Useful Life (RUL) estimation, is used for enhancing the integration of new operational data [2]. Convolutional Neural Networks (CNN) combined with Extreme Gradient Boosting (XGBoost) for fault diagnosis, improving both interpretability and accuracy [3]. More recently, deep learning models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been explored for prognostics of rotating machinery. Found GRU to have faster convergence and robust predictive performance for wind turbine condition monitoring [4]. LSTM is also used to optimize gas turbine maintenance, achieving notable gains in reliability and cost efficiency [5]. Despite these advancements comparative research on LSTM and GRU specifically for

gas turbine performance prediction remains limited. Accordingly, this study develops and compares both models using historical data from an offshore platform to evaluate predictive accuracy and computational efficiency. By leveraging deep learning, we aim to strengthen predictive maintenance strategies, enhance turbine reliability, and minimize operational downtime in offshore energy operations.

2. Material and Methods

This study utilized historical operational data from a gas turbine operating on an offshore oil and gas platform. Data were collected via the Distributed Control System (DCS) and Open Platform Communications (OPC) server from December 2018 to April 2020. Recorded operational parameters included Suction Temperature, Discharge Temperature, Gas Producer Speed, Fuel Flow, and Lube Oil Pressure, with gas turbine shaft power as the target variable.

2.1 Data Preprocessing

Data preprocessing was conducted to ensure data quality, including:

a. Data Cleaning: Removed invalid entries and missing values.
b. Outlier Removal: Applied IQR method to eliminate significant outliers.
c. Normalization: Used min-max scaling (0–1) to enhance model training stability.

2.2 Feature Selection

Two feature selection methods were employed to identify important variables significantly influencing the output:

a. Random Forest Importance: Features were ranked based on importance scores.
b. Least Absolute Shrinkage and Selection Operator (LASSO): A regularization technique that reduces coefficients of less significant variables towards zero.

Selected features for modeling included High Pressure Compressor Flow, Average Exhaust Gas Temperature, High Pressure Compressor Suction Temperature, High Pressure Compressor Discharge Temperature, Fuel Pressure, and High Pressure Compressor Efficiency.

2.3 Development

2.3.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a specialized form of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. This is achieved through the use of memory cells and gating mechanisms. The structure and function of an LSTM cell are described in detail by equations (1) to (6).

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

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

$$\hat{c}_t = tanh(W_c \, x \, [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_f * C_{t-1} + i_t * \hat{c}_t \quad (4)$$

$$O_t = \sigma(W_0 \, x \, [h_{t-1}, x_t] + b_0) \quad (5)$$

$$h_t = O_t \, x \tanh(C_t) \quad (6)$$

Where :

f_t	:	Forget gate
i_t	:	Input gate
$\hat{c}_t$	:	Candidate cell state
C_t	:	Cell output at the current time t
O_t	:	Output gate
h_t	:	Cell output at the current time t
C_{t-1}	:	cell outputs at the previous time x_{t-1}
h_{t-1}	:	cell outputs at the previous time x_{t-1}
x_t	:	Input to the LSTM cell
W	:	Weight of neurons
B	:	Bias for each weight

2.3.2 Gated Recurrent Unit (GRU)

A variant of RNNs with a simpler architecture compared to LSTM, utilizing reset and update gates for learning sequential dependencies. Both models were constructed with two hidden layers, ReLU activation functions, and a final dense output layer. Hyperparameters such as the number of neurons, batch size, and number of epochs were optimized through experimental trials.

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

$$z_t = \sigma(V_{(xz)} \cdot x_t + W_{(hz)} \cdot h_{(t-1)} + p_z) \quad (8)$$

$$c_t = tanh(V_{xc} \cdot x_t + W_{hc} \cdot (r_t * h_{t-1}) + p_c) \quad (9)$$

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

$$\sigma(t) = \frac{1}{(1+e^{(-t)})} \quad (11)$$

$$F(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \quad (12)$$

Where :

r_t : Reset gate.
z_t : Update gate.
c_t : Candidate hidden state.
h_t : Final hidden state.
σ : Sigmoid activation function.
$tanh$: Activation function.

2.4 Data Splitting

The dataset was randomly divided into:

a. Training set 70%
b. Validation set 15%
c. Testing set 15%

This split ensured balanced data representation across subsets.

2.5 Model Evaluation

To evaluate how well the model performed, several statistical indicators were employed:

a. Mean Squared Error (MSE)
b. Root Mean Squared Error (RMSE)
c. Mean Absolute Error (MAE)
d. Coefficient of Determination (R^2)

These evaluation criteria offered a thorough insight into the predictive accuracy of the model when applied to gas turbine operational datasets.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (13)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (14)$$

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

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (16)$$

Where :

y_i : Represents the actual value of the target variable (measured electrical power).
$\hat{y}_i$: Denotes the predicted value of the target variable (predicted electrical power).
$\bar{y}$: Refers to the mean value of the target variable (average electrical power).
n : Total number of observations used in the prediction.

These metrics provide a comprehensive assessment of the predictive accuracy and robustness of the models.

3. Results and Discussions

3.1 Data Pre-processing

The dataset utilized in this study consisted of operational parameters of a gas turbine system under normal operating conditions collected from June 2018 to April 2020. These parameters captured various operational states and load variations, providing a comprehensive foundation for model development.

3.1.1 Missing Value Analysis and Cleaning

The initial data cleaning step involved removing invalid string entries such as "No Data," "Bad," "Not Connect," and "I/O Timeout," which indicated system errors during data collection. Since these entries did not contain valid numerical values, they were excluded to maintain dataset integrity. Subsequently, missing values across multiple numerical features were addressed by removing incomplete rows. A duplicate check was then performed, confirming that no duplicate entries were present.The dataset was thus cleaned and prepared for the next stages of analysis. A summary of the cleaning process is presented in Table 1.

Table 1. Summary of Missing Record Removal

Description	Data	Remarks
Before Missing Value Removal	1,002,224	Rows
After Missing Value Removal	697,061	Rows
Total Missing Value	305,163	Rows
After Duplicate Check	697,061	Rows

3.1.2 *Analysis Outlier*

Outliers are data points that differ significantly from the majority and can distort statistical analysis and reduce machine learning accuracy. To maintain data quality, this study detected outliers by analyzing skewness and kurtosis to assess distribution asymmetry and peakedness. Based on the distribution characteristics, the appropriate handling method was selected. For most variables, the Interquartile Range (IQR) method was applied, as the data were approximately normal or slightly skewed. IQR was chosen for its effectiveness in identifying extreme values without high sensitivity to distribution shape. A summary of the outlier analysis is shown in Table 2.

Table 2. Statistics and Distribution Analysis for Outlier Detection

Variable	Count	Mean	Std Dev	Min	25%	50%	75%	Max	Skewness	Kurtosis
ngp	697,061	96.75	9.61	0	96.87	97.90	98.64	100.01	-9.77	95.30
npt	697,061	71.54	7.16	0	71.48	72.35	72.99	78.57	-9.59	92.87
T1	697,061	84.44	6.17	0	83.10	84.93	86.74	97.7	-8.23	87.82
T5 avg	697,061	1,163.13	135.58	0	1,160	1,180	1,205.98	1,254	-6.93	50.34
Lube oil pressure	697,061	48.36	6.91	-25	48.31	49.18	50.06	57.62	-7.07	52.30
Lube oil temperature	697,061	132.70	5.15	0	131.39	133.22	134.88	143.76	-7.24	67.59
Mfcv position	697,061	66.71	9.56	-25	65.81	67.78	69.87	80.95	-6.10	41.86
Fuel pressure	697,061	175.33	19.12	-75	177.30	177.50	177.70	212.24	-9.52	94.09
Surge margin	697,061	79.97	22.92	-100	73.50	79.83	88.83	859.95	-1.85	60.08
Hpc suction pressure	697,061	580.07	56.75	-271.75	582.59	586.62	587.81	1,071.5	-10.02	99.79
Hpc discharge pressure	697,061	1,124.26	110.64	-543.75	1,128.07	1,135.61	1,142.71	1,308.8	-9.90	97.46

Hpc suction temperature	697,061	77.99	2.35	0	76.80	78.00	79	115.33	5.32	88.77
Hpc disch temperature	697,061	197.84	11.08	0	197.34	198.89	200.23	247.60	-9.09	90.82
Hpc flow	697,061	16.34	2.49	0	14.40	17.30	18	23.01	-2.95	14.75
Hpc head	697,061	29.25	1.97	0	29.46	29.46	29.46	29.46	-10.99	127.28
Hpc efficiency	697,061	0.62	0.86	-31.85	0.63	0.635	0.64	31.88	-0.41	762.18
Engine power	697,061	905.49	171.64	0	748.77	969.50	1,060	1,266.90	-1.85	6.77

3.1.3 Outlier Detection and Removal

Outlier removal was performed using the Interquartile Range (IQR) method. For each variable, Q1 and Q3 were calculated, and data falling below Q1 − 1.5×IQR or above Q3 + 1.5×IQR were identified as outliers and excluded from the dataset. The detailed results of the outlier detection process, including the calculated quartiles, bounds, and the number of removed observations for each feature, are summarized in Table 3.

Table 3. Summary Outlier Feature

Variable	Q1	Q3	IQR	Lower Bound	Upper Bound	Data Removal
Ngp	96.88	98.65	1.77	94.23	101.3	15,166
Npt	71.53	73.02	1.49	69.3	75.25	13,283
T1	83.17	86.76	3.59	77.78	92.15	13,574
T5 avg	1162	1206	44	1096	1272	291
Lube oil pressure	48.38	50.06	1.69	45.85	52.59	630
Lube oil temperature	131.5	134.87	3.37	126.45	139.92	1,120
Mfcv position	66.06	70	3.94	60.14	75.91	2,329
Fuel pressure	177.31	177.69	0.38	176.73	178.27	19,581
Surge margin	74.37	88.8	14.44	52.71	110.46	6,662
Hpc suction pressure	582.59	587.75	5.16	574.86	595.48	9,984
Hpc discharge pressure	1,129.18	1,142.71	13.53	1,108.89	1,163.01	3,955
Hpc suction temperature	76.77	78.9	2.13	73.57	82.1	1,979
Hpc discharge temperature	197.58	200.24	2.67	193.58	204.24	10,285
Hpc flow	14.4	18.07	3.67	8.9	23.57	0
Hpc head	29.46	29.46	0	29.46	29.46	21
Hpc efficiency	0.63	0.64	0.01	0.62	0.66	4,091
Engine power	752	1,060	308	290	1,522	0

Table 4. Summary of Data Cleaning Process

Description	Data	Remarks

Before Missing value Removal	1,0002,224	Rows
After Missing value Removal	697,061	Rows
After Duplicate Check	697,061	Rows
After Outlier Removal	594,110	Rows

A summary of the data cleaning process is presented in Table 4. After initial data collection, the dataset underwent a series of cleaning steps to improve data quality. First, string-type missing values were removed, followed by a duplicate check to ensure data consistency. Afterward, outlier detection and removal were conducted to eliminate extreme deviations.The final clean dataset is now ready for use in the subsequent analysis.

Figure 1. illustrates the comparison between the dataset before and after outlier removal. The red line represents the original data, which contains several extreme deviations and irregular spikes in Engine Power over time. After applying the outlier removal process, shown by the blue line, the data becomes significantly smoother and more consistent, indicating a cleaner and more reliable dataset for further analysis.

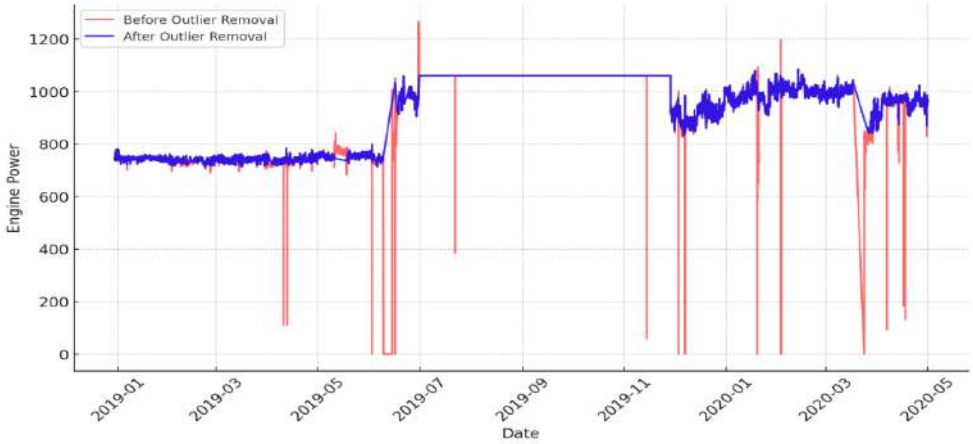

Figure 1. Comparison Of Engine Power Over Time Before vs After Outlier Removal

3.2 *Feature Selection*

Feature selection was performed using Random Forest and LASSO regression to identify the most relevant variables for shaft power prediction. Both methods highlighted similar key features, including HPC Flow, Exhaust Gas Temperature, HPC Suction and Discharge Temperature, Fuel Pressure, and HPC Efficiency. These features improved model accuracy and reduced overfitting.

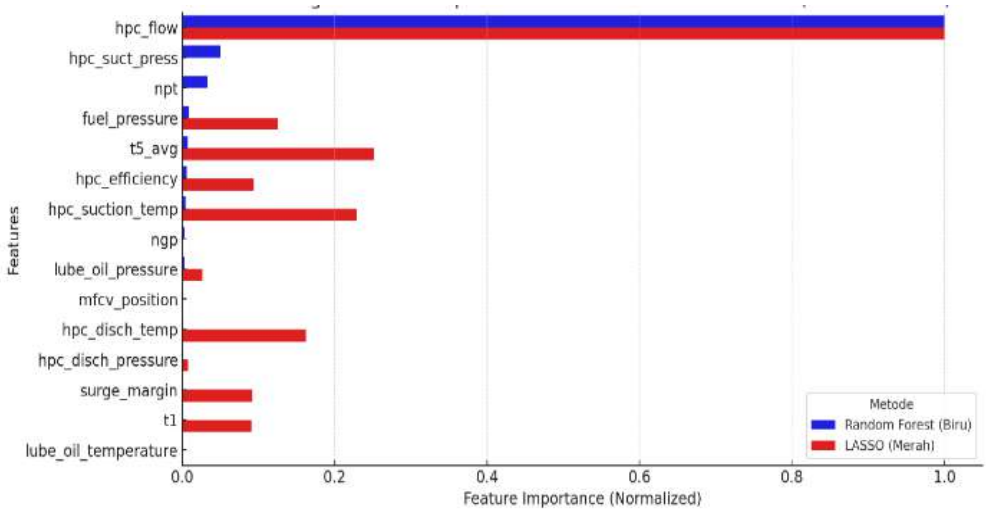

Figure 2. Comparison Of Random Forest vs Least Absolute Shrinkage and Selection Operator

3.3 Data Normalization

After outlier removal, Min-Max normalization was applied to rescale all feature values within the [0, 1] range. This normalization procedure standardizes feature magnitudes, facilitating faster convergence and improved stability during model training.

3.4 Data Splitting

The normalized dataset was partitioned into three subsets:

a. 70% for model training
b. 15% for validation
c. 15% for testing

The dataset was partitioned into three distinct subsets to support effective model training and evaluation. Specifically, the training set was used to fit the model parameters, the validation set was employed for hyperparameter tuning and overfitting monitoring, and the test set was reserved for assessing the model's generalization capability on previously unseen data. The detailed distribution of the data split is summarized in Table 5.

Table 5. Dataset Splitting Results

Dataset	Numbers of Samples	Data Dimensions
Training	415,877	(415,877, 6)
Validation	89,116	(89,116, 6)
Testing	89,117	(89,117, 6)

3.5 Model Architecture

Figure 3 shows the architecture of the deep learning model used for shaft power prediction, applicable to both LSTM and GRU configurations. The model starts with an Input Layer receiving six features: hpc flow, ts avg, hpc suction temp, hpc disch temp, fuel pressure, and hpc efficiency. These are processed through two recurrent layers (LSTM or GRU) with 64 and 32 neurons, each followed by a 20% Dropout layer to prevent overfitting. A Dense Layer with 16 neurons using ReLU activation extracts high-level features, and a final Output Layer with one neuron predicts engine power. The architectures are identical, differing only in the type of recurrent unit.

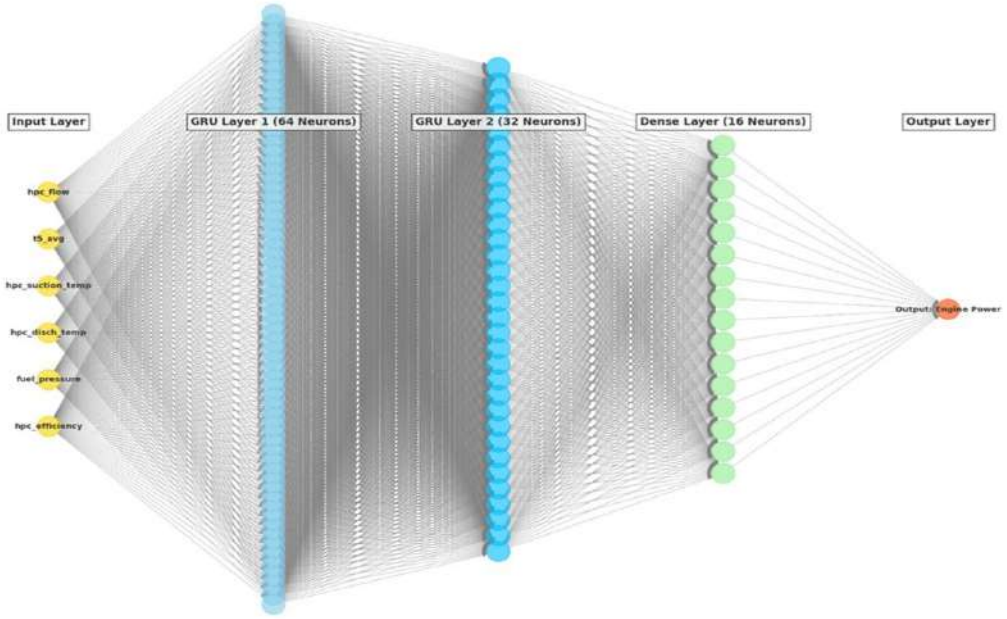

Figure 3. Model Architecture for LSTM and GRU Network

3.5.1 Model Training

Based on Table 4, LSTM and GRU models were trained using Adam optimizer (learning rate = 0.001) and Mean Squared Error (MSE) as the loss function. Experiments varied batch sizes (32, 64, 128) and epochs (100, 200, 300), resulting in eighteen training runs. The choice of these parameters aimed to observe model learning dynamics and convergence behavior, ensuring fair comparisons by keeping other hyperparameters constant.

Table 4. Configuration of Training Parameters for LSTM and GRU Models

No	Loss Function	Batch size	Epochs	Learning Rate	Model	Optimizer
1	MSE	32	100	0.001	LSTM	Adam
2	MSE	32	200	0.001	LSTM	Adam
3	MSE	32	300	0.001	LSTM	Adam
4	MSE	64	100	0.001	LSTM	Adam
5	MSE	64	200	0.001	LSTM	Adam
6	MSE	64	300	0.001	LSTM	Adam
7	MSE	128	100	0.001	LSTM	Adam
8	MSE	128	200	0.001	LSTM	Adam
9	MSE	128	300	0.001	LSTM	Adam
10	MSE	32	100	0.001	GRU	Adam
11	MSE	32	200	0.001	GRU	Adam
12	MSE	32	300	0.001	GRU	Adam
13	MSE	64	100	0.001	GRU	Adam
14	MSE	64	200	0.001	GRU	Adam

15	MSE	64	300	0.001	GRU	Adam
16	MSE	128	100	0.001	GRU	Adam
17	MSE	128	200	0.001	GRU	Adam
18	MSE	128	300	0.001	GRU	Adam

3.6 Training Time Efficiency Analysis

In addition to evaluating predictive accuracy, this study also analyzed the training time efficiency of the LSTM and GRU models. Training durations were measured across various combinations of batch sizes and epochs, and the results are visualized in Figure 4.

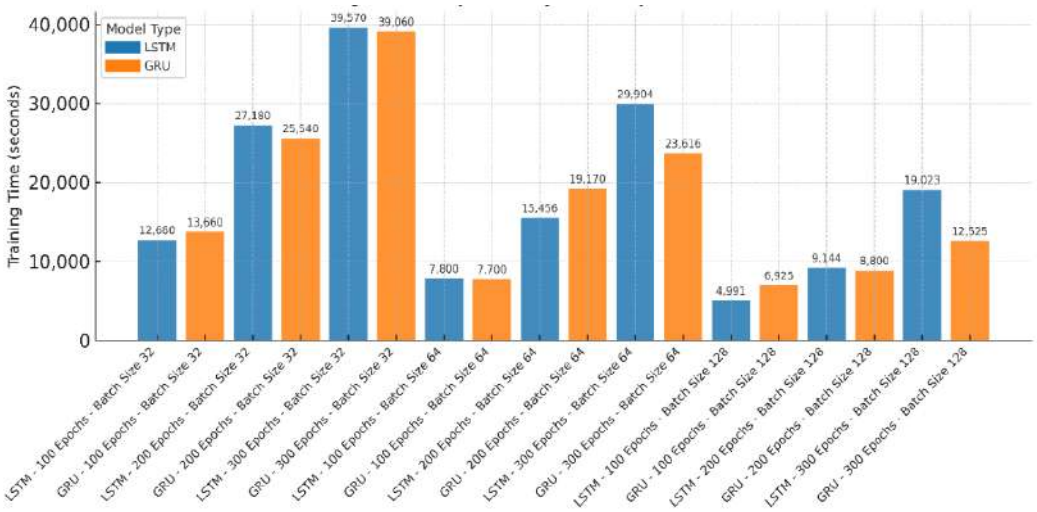

Figure 4 Training Time Comparison by Model, Epochs and Batch size

The analysis indicates that GRU generally requires less training time compared to LSTM, especially in configurations with larger batch sizes and higher epoch counts. However, in certain cases—such as with a batch size of 32 and 100 epochs—GRU required slightly more time than LSTM. This suggests that while GRU is not consistently faster in every scenario, it still demonstrates superior computational efficiency in most training conditions.

3.7 Baseline Model Performance

To establish a performance benchmark, a Linear Regression model was developed using the same training dataset and selected features as the LSTM and GRU models. The goal was to evaluate the improvement offered by deep learning models over a conventional approach.

Table 6. Baseline Linear Regression Model Performance

Metric	Value
Mean Squared Error (MSE)	828.5864
Root Mean Squared Error (RMSE)	28.7852
Mean Absolute Error (MAE)	20.1905
Coefficient of Determination (R^2 Score)	0.9533

The Linear Regression model achieved an R^2 of 0.9533, indicating reasonable predictive power. However, the higher error values—particularly RMSE—highlight its limitations

in capturing the nonlinear and sequential patterns of gas turbine data. The next section presents a comparative evaluation of LSTM and GRU performance.

3.8 *Model Evaluation*

To evaluate the effectiveness of the LSTM and GRU models, performance was compared using MSE, RMSE, MAE, and R^2 metrics. The optimal LSTM configuration (batch size = 32, epochs = 200) achieved an R^2 of 0.9991, RMSE of 3.96, and MAE of 1.83, while the best GRU configuration (batch size = 32, epochs = 300) reached an R^2 of 0.9988 and RMSE of 4.58. Figure 4 illustrate comparisons of R^2 and MSE scores, respectively, demonstrating that LSTM consistently provides higher accuracy with lower errors than GRU. Overall, LSTM offers superior predictive accuracy, making it preferable for applications requiring high precision, whereas GRU is advantageous for faster training and computational efficiency.

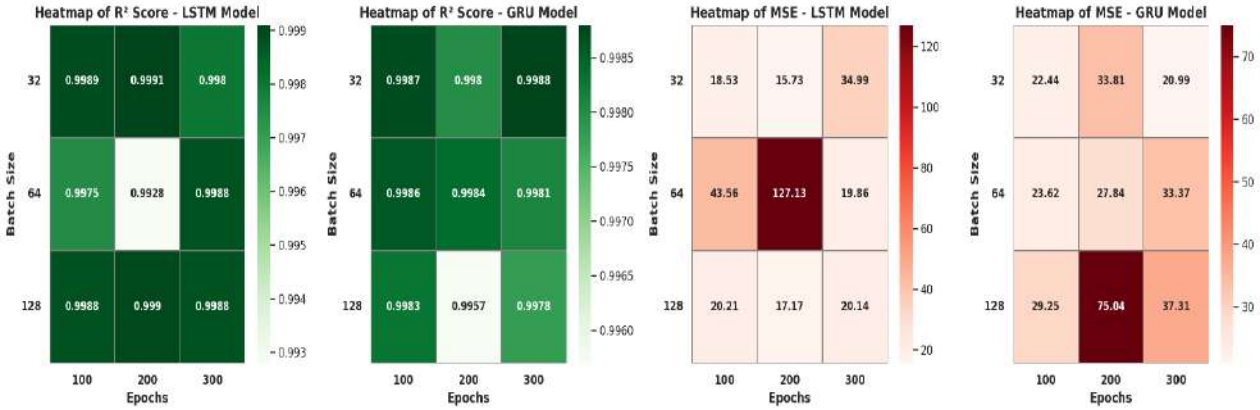

Figure 4. Heatmap Comparison of R^2, MSE Score for LSTM and GRU Model

3.9 Testing Results and Error Analysis

After completing the training and evaluation phases, the final step was to test the models using an unseen testing dataset. This stage was designed to assess the models' ability to generalize to new data. The best configurations for each architecture LSTM and GRU, both with a batch size of 32 and 200 epochs were selected for the testing process. The testing results were obtained by comparing the predicted outputs of both models against the actual test data. As shown in Figure 5, the prediction curve produced by the LSTM model aligns more closely with the actual data pattern than the GRU model, indicating superior predictive performance.

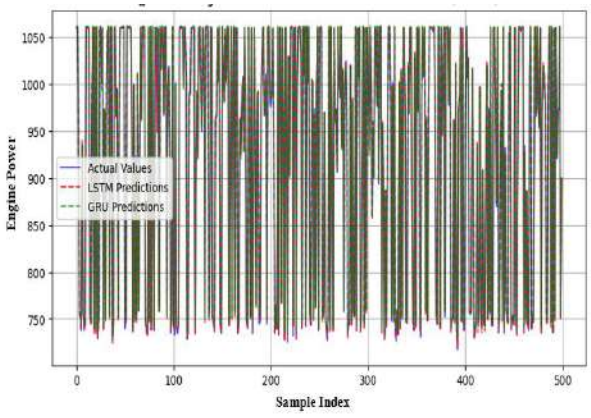

Figure 5. Testing Comparison: LSTM and GRU Models vs. Actual Data

Furthermore, Figure 6 provides a more detailed illustration of the forecasting results for both models using their optimal configurations.

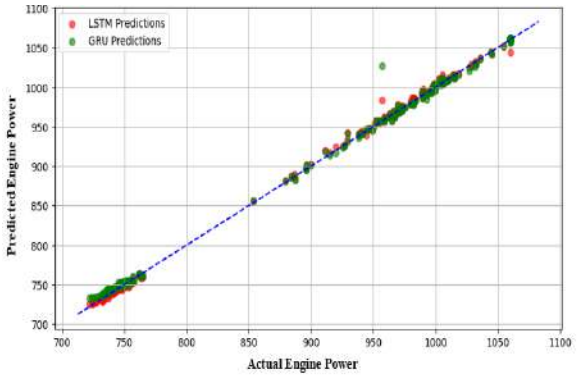

Figure 6. Forecasting Results of the LSTM and GRU Models (Batch Size = 32, Epochs = 200).

In addition, Figure 7 presents the histogram of prediction errors for both models. The LSTM model exhibits a narrower and more concentrated error distribution around zero, indicating smaller and more consistent prediction errors compared to the GRU model.

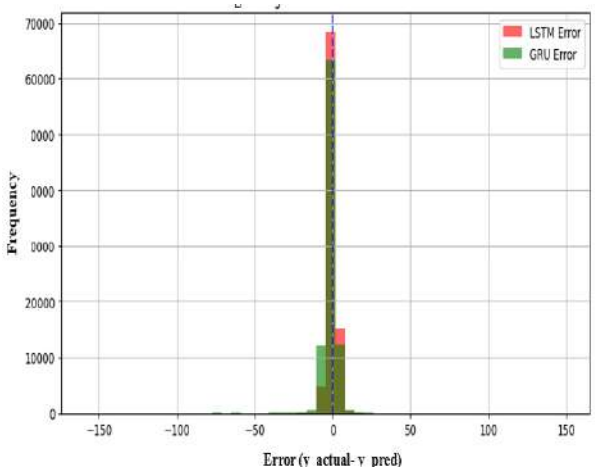

Figure 7. Histogram of Prediction Errors for the LSTM and GRU Models (Batch Size = 32, Epochs = 200)

Overall, these testing results further reinforce the previous findings, confirming that the LSTM model achieves superior predictive performance in forecasting gas turbine shaft power output. While the GRU model offers greater computational efficiency, the LSTM model demonstrates higher accuracy and reliability in time-series prediction tasks.

4. Conclusion

This study evaluated LSTM and GRU models for gas turbine performance prediction using historical operational data from an offshore platform. The data preprocessing steps including cleaning, feature selection, and normalization successfully improved data quality for model training. The LSTM model achieved the best predictive performance with an RMSE of 3.96 and an R^2 of 0.9991, slightly outperforming the GRU model (RMSE 4.58; R^2 0.9988). In comparison, the baseline linear regression model achieved only R^2 0.9533 and RMSE 28.78, highlighting the superior capability of deep learning models in capturing nonlinear and temporal patterns.

These findings support the implementation of the LSTM model in gas lift systems, where gas turbines drive compressor packages to maintain reservoir pressure. By integrating the model into SCADA or DCS systems, operators can monitor shaft power performance in real time, detect early degradation, and plan preventive maintenance. This approach enhances operational reliability and energy efficiency in dynamic offshore environments.

Reference

[1] Liu Z, Karimi IA. Gas turbine performance prediction via machine learning. *Energy*, 2020; 192. https://doi.org/10.1016/j.energy.2019.116627

[2] Zaidan MA, Mills AR, Harrison RF, Fleming PJ. Gas turbine engine prognostics using Bayesian hierarchical models: A variational approach. *Mechanical Systems and Signal Processing*, 2016; 70–71:120–140. https://doi.org/10.1016/j.ymssp.2015.09.014

[3] Zhou D, Yao Q, Wu H, Ma S, Zhang H. Fault diagnosis of gas turbine based on partly interpretable

convolutional neural networks. *Energy*, 2020; 200. https://doi.org/10.1016/j.energy.2020.117467
[4] Kerboua A, Kelaiaia R. Recurrent neural network optimization for wind turbine condition prognosis. *Diagnostyka*, 2022; 23(3). https://doi.org/10.29354/diag/151608
[5] Djeddi AZ, Hafaifa A, Hadroug N, Iratni A. Gas turbine availability improvement based on long short-term memory networks using deep learning of their failures data analysis. *Process Safety and Environmental Protection*, 2022; 159:1–25. https://doi.org/10.1016/j.psep.2021.12.050