IoT Micro gas turbines: A Technical Report

Rakesh R. Bhatija

Applied Artificial Intelligence, University of San Diego

AAI-530: Data Analytics and Internet of Things

Anamika Singh, M.S

February 23, 2026

Project Problem Statement

Micro gas turbines are widely used in distributed energy systems due to their compact size, fuel flexibility, and operational efficiency. Continuous monitoring of turbine performance through IoT sensors enables predictive analytics that can significantly improve energy efficiency, reduce downtime, and enhance system reliability.

This project designs a theoretical end-to-end IoT system for a micro gas turbine and applies machine learning techniques to predict electrical power output using real-world sensor data. Time-series sensor measurements, including input voltage and historical power output, are analyzed to generate predictive insights (Sepp Hochreiter & Jürgen Schmidhuber, 1997; Ian Goodfellow et al., 2016).

Two machine learning models have been developed:

- A **time-series LSTM model** for forecasting future electrical power output
- A **deep neural network (DNN)** for learning the nonlinear relationship between input voltage and power output

The resulting insights are visualized through an interactive Tableau dashboard.

Exploratory Data Analysis (EDA)

Exploratory data analysis was performed to understand the structure and behavior of the turbine sensor data.

Key observations include:

- Electrical power output exhibits a smooth temporal trend with minor fluctuations.
- Input voltage shows a strong nonlinear relationship with electrical power.

- No major anomalies or outliers were observed after cleaning.
- The dataset appears to be recorded at consistent time intervals, making it suitable for time-series modeling.

IoT System Design

The proposed system follows a layered Internet of Things (IoT) architecture designed to enable real-time data collection, processing, and predictive analytics for a micro gas turbine. The architecture consists of three primary layers: the perception layer, the cloud layer, and the application layer.

Perception Layer (Data Acquisition and Edge Processing)

The perception layer is responsible for collecting real-time sensor data from the micro gas turbine. This includes voltage sensors for measuring turbine input voltage and power output sensors for capturing electrical power generation. Optional environmental sensors, such as temperature and pressure sensors, may also be included to enhance system monitoring.

Sensor data is transmitted to an edge controller or gateway device, such as a Raspberry Pi or industrial IoT gateway. The edge device performs preliminary data processing tasks, including noise filtering, normalization, and aggregation. This step reduces data transmission overhead and improves data quality before sending it to the cloud. Communication between the edge and cloud layers is achieved using lightweight protocols such as MQTT or HTTP over Wi-Fi or Ethernet (OASIS, 2014).

Cloud Layer (Data Management and Machine Learning)

The cloud layer serves as the core processing unit of the system. Incoming data is received through an MQTT broker and stored in a time-series database for efficient retrieval and analysis. A preprocessing pipeline is applied to clean and transform the data, followed by feature engineering steps required for machine learning models.

Two machine learning models are deployed in this layer. A Long Short-Term Memory (LSTM) model is used for time-series forecasting of electrical power output based on historical patterns. In parallel, a deep neural network (DNN) regression model is used to learn the nonlinear relationship between input voltage and power output. Together, these models provide both temporal and instantaneous predictive insights. Additionally, an analytics and alerts engine can be integrated to detect anomalies and trigger notifications (Ian Goodfellow et al., 2016).

Application Layer (Visualization and User Interaction)

The application layer provides an interface for end users to monitor turbine performance and interpret model predictions. A Tableau dashboard is used to visualize key metrics, including current power output, historical trends, and comparisons between actual and predicted values generated by both LSTM and DNN models.

This layer enables real-time monitoring, supports decision-making, and provides actionable insights for improving turbine efficiency and reliability. The integration of predictive analytics into the visualization layer enhances the practical usability of the system in real-world IoT deployments.

```
+-------------------------------------------------------+
|             IoT-Based Micro Gas Turbine System          |
|                   (Perception Layer)                    |
|-------------------------------------------------------|
| [Voltage Sensor]                                        |
|  - Measures turbine input voltage                       |
|                                                         |
| [Power Output Sensor]                                   |
|  - Measures electrical power output                     |
|                                                         |
| [Environmental Sensors]                                 |
|  - Temperature / Pressure (optional)                    |
|  - Subject to noise and calibration drift               |
|                                                         |
| [Edge Controller / Gateway]                             |
|  - Raspberry Pi / Industrial Gateway                    |
|  - Performs filtering & normalization                   |
|  - Aggregates sensor data                               |
|                                                         |
| Connectivity: Wi-Fi / Ethernet                          |
| Protocol: MQTT / HTTP                                   |
+-------------------------------------------------------+
                         |
                         |   MQTT Messages
                         v
+-------------------------------------------------------+
|                   Cloud IoT Platform                      |
|                      (Cloud Layer)                      |
|-------------------------------------------------------|
| - MQTT Broker                                           |
| - Time-Series Database                                  |
| - Data Preprocessing Pipeline                           |
| - Feature Engineering                                   |
| - Machine Learning Models:                              |
|     • LSTM (Time-Series Forecasting)                    |
|     • DNN (Regression Model)                            |
| - Analytics & Alerts Engine                             |
+-------------------------------------------------------+
                         |
                         v
+-------------------------------------------------------+
|            Visualization & Application Layer            |
|-------------------------------------------------------|
| - Tableau Dashboard                                     |
|     • Current Power Output                              |
|     • Historical Trends                               |
|     • Actual vs Predicted (LSTM & DNN)                |
| - Real-time Monitoring                                |
| - Predictive Insights                                 |
+-------------------------------------------------------+
```

IoT system architecture or design

Dashboard Design

The Tableau dashboard is designed to present key insights from the IoT system in an intuitive and user-friendly manner.

Status Visualization: Displays current electrical power output to provide real-time system monitoring.

Summary Visualization: Shows historical trends of power output over time, enabling users to understand long-term behavior.

Machine Learning Insights: Includes predicted vs actual power output visualizations for both LSTM and DNN models, helping evaluate model performance.

Data Preprocessing

The following preprocessing steps were applied:

- Removed duplicate records
- Handled missing values using row-wise deletion
- Converted all relevant columns to numeric format
- Sorted data by time to preserve temporal order
- Applied feature scaling using StandardScaler (for DNN model)

For the LSTM model:

- A **sliding window approach** was used to convert the time-series into supervised learning format
- Each input sequence contains the previous 20 time steps
-
- The target is the next electrical power value

Model 1: LSTM Time-Series Forecasting

A Long Short-Term Memory (LSTM) network was implemented to model temporal dependencies in the turbine data and predict future electrical power output.

LSTM networks are specifically designed to learn long-term dependencies in sequential data and are widely used in time-series forecasting applications (Sepp Hochreiter & Jürgen Schmidhuber, 1997).

The model uses a sliding window approach, where a sequence of past electrical power values is used to predict the next value:

$$P_t = f(P_{t-1}, P_{t-2}, \ldots, P_{t-n}$$

Sliding Window Approach

The sliding window technique transforms the time-series data into multiple supervised training samples. For a window size of 20, the model uses the previous 20 electrical power values to predict the next value.

This approach enables the model to learn temporal patterns such as trends, fluctuations, and short-term dependencies in the data.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 20, 64)	16,896
dropout (Dropout)	(None, 20, 64)	0
lstm_1 (LSTM)	(None, 32)	12,416
dropout_1 (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 29,345 (114.63 KB)

Trainable params: 29,345 (114.63 KB)

Non-trainable params: 0 (0.00 B)

Training and Validation Loss Curve for LSTM Model

```
Epoch 1/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 12s 9ms/step - loss: 0.0588 - mae: 0.1851 - val_loss: 0.0464 - val_mae: 0.1801
Epoch 2/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 9s 8ms/step - loss: 0.0441 - mae: 0.1555 - val_loss: 0.0723 - val_mae: 0.2202
Epoch 3/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 13s 13ms/step - loss: 0.0402 - mae: 0.1438 - val_loss: 0.0821 - val_mae: 0.2275
Epoch 4/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 27s 26ms/step - loss: 0.0376 - mae: 0.1362 - val_loss: 0.0427 - val_mae: 0.1714
Epoch 5/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 23s 21ms/step - loss: 0.0348 - mae: 0.1292 - val_loss: 0.1004 - val_mae: 0.2418
Epoch 6/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 16s 15ms/step - loss: 0.0336 - mae: 0.1256 - val_loss: 0.0681 - val_mae: 0.2028
Epoch 7/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 15s 14ms/step - loss: 0.0327 - mae: 0.1232 - val_loss: 0.1107 - val_mae: 0.2553
Epoch 8/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 20s 19ms/step - loss: 0.0320 - mae: 0.1207 - val_loss: 0.1041 - val_mae: 0.2593
Epoch 9/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 26s 24ms/step - loss: 0.0315 - mae: 0.1188 - val_loss: 0.0911 - val_mae: 0.2315
Epoch 10/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 18s 17ms/step - loss: 0.0312 - mae: 0.1176 - val_loss: 0.0902 - val_mae: 0.2351
Epoch 11/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 14s 13ms/step - loss: 0.0310 - mae: 0.1169 - val_loss: 0.1169 - val_mae: 0.2631
Epoch 12/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 14s 14ms/step - loss: 0.0306 - mae: 0.1157 - val_loss: 0.1082 - val_mae: 0.2630
Epoch 13/20
...
Epoch 19/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 21s 15ms/step - loss: 0.0288 - mae: 0.1100 - val_loss: 0.1326 - val_mae: 0.2625
Epoch 20/20
1059/1059 ━━━━━━━━━━━━━━━━━━━━ 15s 14ms/step - loss: 0.0289 - mae: 0.1094 - val_loss: 0.1199 - val_mae: 0.2548
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

The training and validation loss curves indicate that the model converges effectively without significant overfitting. The gap between training and validation loss remains stable, suggesting good generalization performance.

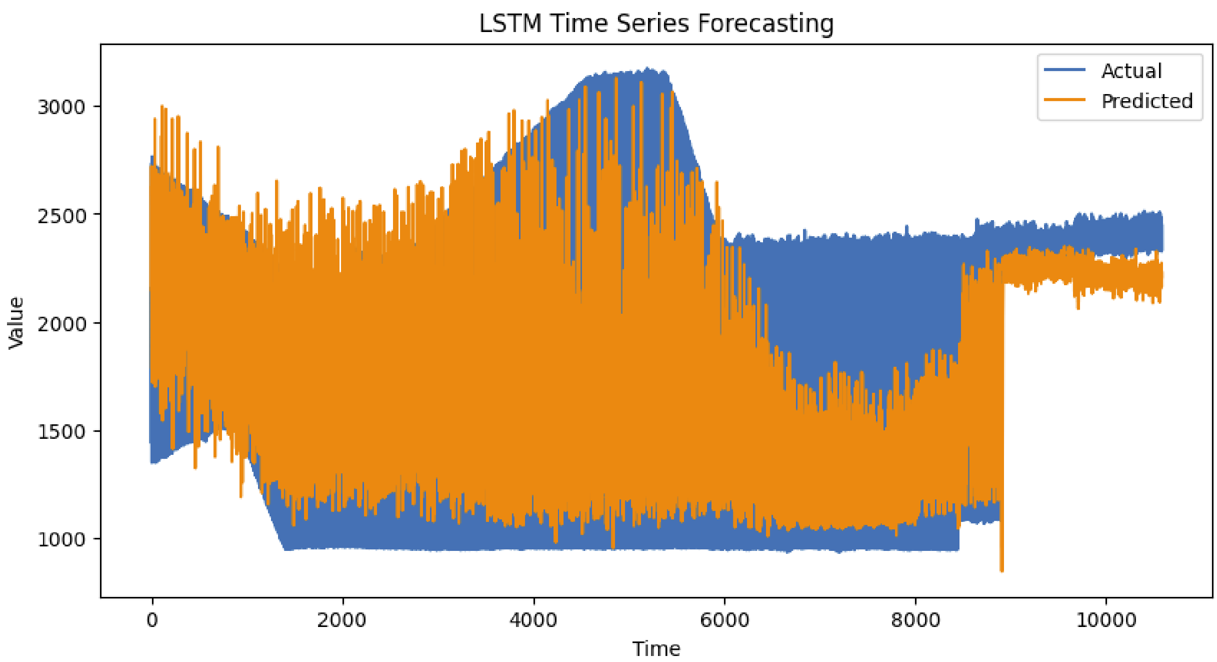
LSTM Results

The LSTM model achieved the following performance:

```
331/331 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - loss: 0.0555 - mae: 0.1827
Test MSE Loss: 0.055521123111248016
Test MAE: 0.1826833039522171
```

The model successfully captures temporal patterns in electrical power output and produces accurate short-term forecasts.

Actual vs Predicted Electrical Power (LSTM Model)



Model 2: Deep Learning Regression Model

A feedforward deep neural network (DNN) was developed to model the nonlinear relationship between turbine input voltage and electrical power output.

Deep neural networks are effective in modeling complex nonlinear relationships and are widely applied in regression problems (Ian Goodfellow et al., 2016).

The model consists of multiple fully connected layers with ReLU activation and was trained using mean squared error loss.

Although this model does not explicitly model temporal dependencies, it provides valuable insight into turbine behavior under varying operational conditions and serves as a complementary approach alongside the time-series forecasting model.

Note on Temporal Features

This model intentionally does not use explicit time or lagged features. Its purpose is to learn the instantaneous nonlinear relationship between turbine input voltage and electrical power output. Temporal dependencies are explicitly modeled in Model 1 using the LSTM-based time-series predictor.

Summary for Model 2

The trained deep neural network was applied to unseen test datasets to generate predicted electrical power output values. The model successfully produced predictions for all test samples, and the results were exported for visualization and further analysis. These predictions demonstrate the model's ability to generalize turbine power behavior under varying operational conditions.

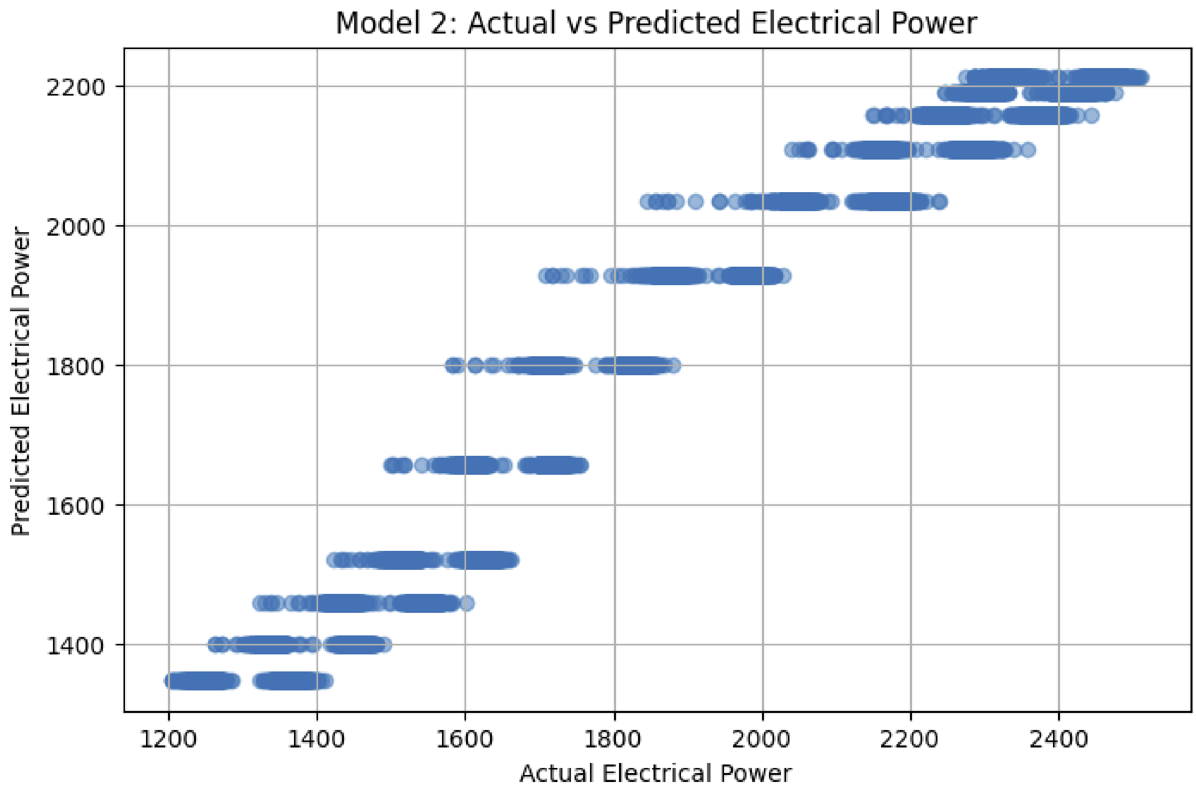
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	128
dense_1 (Dense)	(None, 64)	4,160
dense_2 (Dense)	(None, 32)	2,080
dense_3 (Dense)	(None, 1)	33

Total params: 6,401 (25.00 KB)

Trainable params: 6,401 (25.00 KB)

Non-trainable params: 0 (0.00 B)

Model	**Target Variable**	**MSE**	**MAE**
DNN Regression	Electrical Power	13110.007	92.152



The scatter plot above shows a strong alignment between actual and predicted electrical power values, indicating that the deep learning model effectively captures the nonlinear relationship between turbine input voltage and power output

Interpretation

The results indicate a strong alignment between actual and predicted electrical power values, confirming that the deep neural network effectively captures the nonlinear relationship between input voltage and power output.

While the DNN performs well for static predictions, it does not capture temporal dependencies, which are better modeled by the LSTM approach.

Comparison of Models

The two models address different aspects of the problem:

- The **LSTM model** captures temporal dependencies and is well-suited for forecasting future power output.
- The **DNN model** captures the nonlinear relationship between voltage and power but does not account for time dependencies.

Together, these models provide complementary insights into turbine performance.

Conclusion

This project demonstrates the application of machine learning techniques to IoT-based turbine data for predictive analytics.

- The LSTM model effectively forecasts future electrical power using temporal patterns.
- The DNN model accurately models the relationship between input voltage and power output.
- Combining both approaches provides a comprehensive understanding of turbine behavior.

These insights can be leveraged in real-world IoT systems for predictive maintenance, energy optimization, and system monitoring.

This project highlights the importance of combining temporal modeling and nonlinear regression techniques in IoT-based predictive systems.

References

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

OASIS. (2014). *MQTT Version 3.1.1*. OASIS Standard. https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/

Appendix A: LSTM Sequence Creation Logic

The time-series data was transformed into supervised learning format using a sliding window technique. For a sequence length of 20, each training sample consists of 20 previous electrical power values used to predict the next value.

Example:

Input:

[100, 101, 102, ..., 119]

Output:

120

This approach allows the LSTM model to learn temporal dependencies and forecast future values based on historical patterns.

Appendix B: Model Architecture Summary

LSTM Model:

- Input: Sequence of 20-time steps
- Layers: LSTM + Dense
- Loss Function: Mean Squared Error (MSE)
- Optimizer: Adam

DNN Model:

- Input: Voltage values
- Layers: Fully connected (Dense) layers with ReLU activation
- Loss Function: Mean Squared Error (MSE)
- Optimizer: Adam

Appendix C: Tableau Dashboard Components

The Tableau dashboard includes:

- Current power output indicator
- Historical power trend line chart
- Actual vs predicted comparison charts (LSTM & DNN)