Data Science and Management Project: Maintenance of Naval Power Plants

1. Introduction to the Problem

Naval propulsion plants represent critical, high-value assets where reliability and efficiency are paramount. Modern naval vessels, such as frigates, frequently employ complex propulsion architectures like CODLAG (COmbined Diesel eLectric And Gas), which integrate Gas Turbines (GT) for high-speed operations. The maintenance of these plants constitutes a significant portion of a vessel's total operational lifecycle costs.

The specific problem addressed in this project concerns the performance decay of Gas Turbine components, specifically the *Compressor* and the *Turbine*. Over time, these components are subject to degradation phenomena—such as fouling, erosion, and corrosion—which reduce thermodynamic efficiency, increase fuel consumption, and raise exhaust temperatures. Since direct physical inspection is impossible during continuous operation, there is a critical need to estimate the health status of these components indirectly, using sensor data (e.g., temperatures, pressures, torques).

The dataset analyzed in this project was generated from a high-fidelity numerical simulator of a naval vessel's gas turbine [1]. It includes 11,934 observations of 16 sensor measurements and 2 decay coefficients, simulating various degradation scenarios over time.

2. Description of the State of the Art

Traditional maintenance strategies in the naval sector have historically relied on Corrective Maintenance (repairing after failure) or Preventive Maintenance (scheduled replacements regardless of actual condition). However, these approaches are often inefficient: they can lead to catastrophic failures at sea or, conversely, to unnecessary costs when replacing healthy components [2].

The current state of the art is shifting towards Condition-Based Maintenance (CBM) and Predictive Maintenance (PdM). Recent studies demonstrate that applying Machine Learning to sensor data allows for the early detection of anomalies and the estimation of Remaining Useful Life (RUL) [3]. Unlike previous works that focus solely on binary classification (broken/working), this project aims to quantify the *degree* of decay using regression techniques, providing a more granular insight into the plant's health.

3. Proposed Approach

3.1 Data Ingestion and Exploration Analysis

To address the initial requirements of the assignment, we established a data processing pipeline using the Pandas library. The first challenge concerned the raw dataset (navalplantmaintenance.csv), which lacked a header row and used irregular whitespace as a separator. We implemented a custom ingestion script using the read_csv function with a regular expression separator (sep=r'\s+') to correctly parse the file.

Since the raw data contained anonymous numerical values, we manually mapped the 18 columns to their physical meanings (e.g., assigning lp to Lever Position and v to Ship Speed) based on the official documentation. This step was useful to ensure semantic clarity before any analysis.

Before analyzing the signal distributions, we implemented a structural validation step. We explicitly checked the dataset dimensions, data types, and presence of missing values (NaN) to verify the integrity of the ingestion process.

Subsequently, we performed a statistical profiling using the describe() method to compute key metrics such as mean, standard deviation, and interquartile ranges. Finally, to visually characterize the vessel's behavior, we utilized Matplotlib to generate exploratory plots, specifically focusing on operational regimes, control logic consistency, and thermodynamic efficiency.

3.2 Data Consistency and Outlier Detection Strategy

We implemented a validation pipeline using Python to ensure the reliability of the dataset before model training. Given the critical nature of naval propulsion systems, we adopted a hybrid approach combining statistical analysis with domain-specific physical constraints derived from the official dataset documentation [1] [2].

To address the heterogeneity of the sensor scales identified in the exploration phase, we utilized the Pandas library to perform a Min-Max Normalization. This was implemented using vectorized operations on the DataFrame to map all variables into a dimensionless range [0, 1]. This transformation allowed us to visually compare the distributions of variables with vastly different units using a single normalized boxplot generated via Matplotlib.

Subsequently, we developed two distinct outlier detection mechanisms within the script:

1. Statistical Method (IQR)**:** We utilized the .quantile() method from Pandas to calculate the Interquartile Range (IQR). The script automatically flags data points falling outside the range $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$. This method allows for the identification of values that deviate significantly from the statistical distribution.
2. Physical Domain Check**:** Recognizing that statistical outliers in a mechanical system might represent legitimate high-load operational states, we implemented a custom "Physical Range Check". We defined validity intervals based on the standard naval gas turbine specifications [3]. The code iterates through these domain-specific rules using boolean masking to distinguish between a sensor fault and an extreme maneuver.

3.3 Database Architecture and Implementation

First, we designed a Dimensional Fact Model by grouping attributes according to the ship's components. We identified the main entity "Ship" and four specific components: Propeller, GT, GT Compressor, and High Pressure.

General attributes like "Lever position", "Fuel Flow", "Ship speed", "Turbine Injection Control", and "GGn" were assigned to the main Ship table. The Propeller class contains torque measurements for both Port and Starboard. The GT Compressor class includes inlet/outlet temperatures and

pressures, along with its decay coefficient. Similarly, the GT class holds torque, RPM, exhaust pressure, and its decay coefficient. Finally, High Pressure contains the turbine exit temperature and pressure.

We formalized this structure into a Star Schema. The Fact table contains the main measurements and the foreign keys (Propeller_id, GT_Compressor_id, High_Pressure_id, GT_id) to link the dimension tables. In the Star Model diagram, primary keys were underlined to identify unique records.

To store the data, we implemented a MySQL 8.0 database/datawarehouse. We first created a master table named 'dati' to hold the raw CSV data. Then, we defined the normalized tables (Ship, Propeller, GT_Compressor, GT, High Pressure) specifying data types (e.g., float) and constraints (not null).

Using PyCharm, we loaded the CSV into the 'dati' table via the code df.to_sql(..., if_exists='replace', ...), which automatically handled table recreation. Finally, we populated the specific tables by mapping data from 'dati', renaming columns where necessary to ensure consistency. The data is now fully structured within the relational database.

3.4 Temporal Decay Analysis

Following the data structuring phase, we implemented a specific Python procedure to analyze the temporal behavior of the target variables. The analysis focused on the two key performance indicators extracted from the dataset:

- GT Compressor decay state coefficient (GT1)
- GT Turbine decay state coefficient (GT2)

Implementation Details The analysis was performed using the Pandas library to handle the dataset manipulation and Matplotlib for the visualization. In our script, we treated the dataset index as a proxy for the simulation time step. We generated a dual-axis line plot (`plt.subplots`) to visualize the trajectory of both coefficients simultaneously. This coding approach allowed us to inspect the high-frequency variations of the decay parameters and verify the nature of the simulation (continuous vs. discrete), which is a critical step before selecting the regression algorithms in Point 5 [1].

3.5 Predictive Modeling Strategy

The objective of this phase was to predict the health state of the two critical components: the Compressor (GT1) and the Turbine (GT2), based on the sensor readings.

First, we prepared the dataset. An important step was the removal of both decay coefficients from the input features (X). By removing both, we ensured the models rely strictly on the 16 operational sensors (Temperatures, Pressures, Torque, etc.).

Model 1: Linear Regression - As a first approach, we used the Linear Regression method. We divided the dataset into a train set (70%) and a test set (30%).

- With the Python code model3.fit(X3_train, y_train), the algorithm learns the rules connecting the sensors to the decay.

- With the code y3_pred = model3.predict(X3_test), the algorithm applies learned rules to predict the hidden data.
- The code print(mean_absolute_error(y_test, y3_pred)) allows us to quantify how much the model has made a mistake by calculating the mean absolute error between predicted and real values.

Model 2: Random Forest - To validate the results, we also implemented a Random Forest Regressor. This ensemble method was trained on the same data split to ensure a fair comparison.

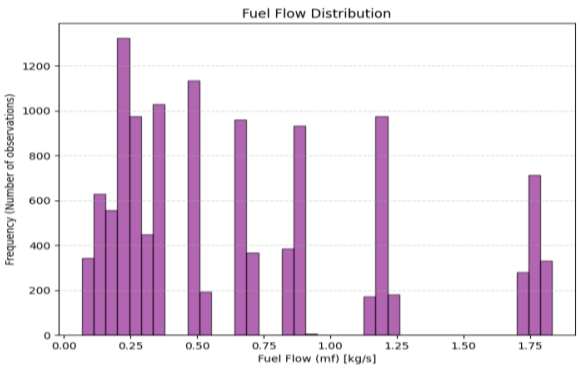
4. Results

4.1 Exploratory Analysis

The preliminary analysis performed via Python allowed us to explore the dataset structure and verify its physical consistency before applying any predictive model. The results confirm the high quality of the simulation data.

The structural checks confirmed that the dataset consists of 11,934 samples and 18 features, all correctly loaded as numerical types (float64). The missing value check returned a zero count for all variables, confirming the high quality of the simulation data and eliminating the need for imputation. The descriptive statistics (df.describe()) revealed that variables exhibit significantly different scales.

The histogram of the Fuel Flow (mf) provides insight into how the vessel was operated during the data collection period.



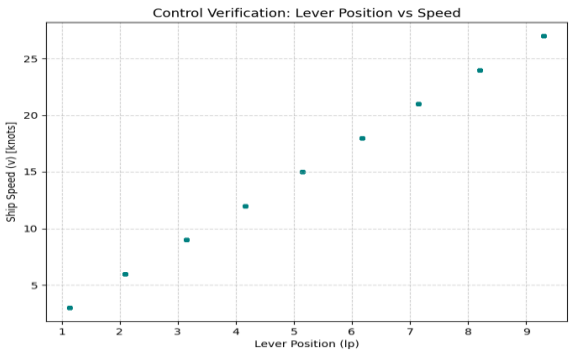
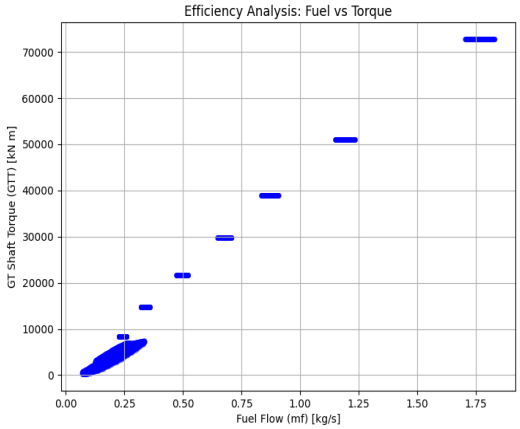


Fig. 1: Distribution of Fuel Flow (mf).

Fig. 2: Lever Position (lp) vs. Ship Speed (v).

As illustrated in Fig. 1, the distribution is not uniform (non-Gaussian). We observe distinct clusters of frequencies. This indicates that the simulation covered specific operational regimes, likely corresponding to standard naval maneuvers (e.g., low-speed patrolling vs. high-speed transit). Understanding these regimes is useful, as gas turbine degradation rates typically vary depending on the engine load.

Fig. 2 shows a strictly monotonic and tight correlation between the *Lever Position* and the *Ship Speed*. This result confirms that the dataset represents a deterministic system with a properly functioning control loop. There are no visible outliers in this relationship, implying that the sensor readings are reliable and consistent with the expected hydrodynamics.



Finally, we investigated the core energy conversion process of the Gas Turbine to rule out simulation errors.

As shown in Fig. 3, the relationship between fuel input and torque output is linear. This confirms that the laws of thermodynamics are respected: higher energy input translates directly into mechanical power, validating the dataset for the decay analysis.

Fig. 3: Fuel Flow (mf) vs. Shaft Torque (GTT).

4.2 Consistency Analysis Results

The execution of the consistency and outlier detection pipeline confirmed the high fidelity of the simulation data. As illustrated in Figure 4, the normalization of the 18 variables onto a [0, 1] scale allows for a direct comparison of their variance.

We observe two distinct behaviors:

1. Constant Parameters: The variables T1 (GT Compressor inlet air temperature) and P1 (GT Compressor inlet air pressure) appear as collapsed boxplots with near-zero variance. This confirms the descriptive statistics from Point 1, indicating that the simulation was performed under constant ambient conditions [2].

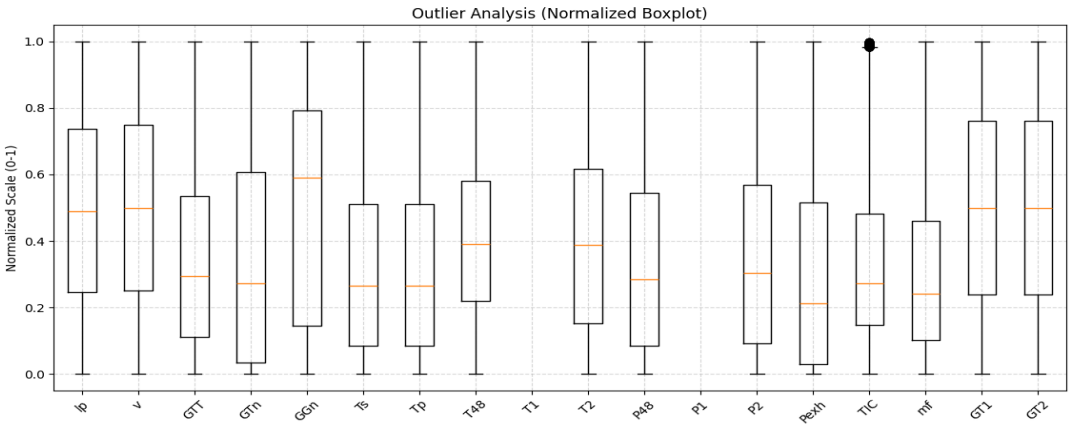


Figure 4: Outlier Analysis (Normalized Boxplot).

2. Dynamic Variables: The remaining mechanical and thermodynamic variables (e.g., GTT, GTn, T48) exhibit a wide spread across the normalized range, indicating a rich dataset that covers various operational regimes.

The boxplot visualization highlights the presence of statistical outliers (black points), particularly evident in the TIC (Turbine Injection Control) variable, where values cluster near the upper bound (1.0). According to the IQR method, these values are considered anomalies.

- Interpretation: The "outliers" visible in Figure 4 for TIC and other variables correspond to valid high-transient maneuvers or maximum power states, not sensor malfunctions.

- Decision: Consistent with the approach taken in the reference Mobility Project—where outliers representing special events were retained—we decided not to remove any records. Removing these high-load data points would remove the most critical information needed to train the regression model on engine decay.

To validate this result, we benchmarked the dataset values against the technical datasheet of the General Electric LM2500, the industry-standard gas turbine for Frigates [3]. The maximum Torque observed in our analysis (~72.7 kNm) aligns perfectly with the LM2500 nominal performance at 3600 rpm [3]. Furthermore, the behavior of the dynamic variables is consistent with the operation of CODLAG propulsion systems, where the gas turbine is engaged specifically for high-speed phases, creating necessary transients that statistical methods misinterpret as outliers [4] [5].

4.3 Database Implementation and Integrity Verification

The data warehousing phase was completed successfully, resulting in the migration of the entire simulation dataset from the raw flat-file format into a structured MySQL relational database named naval_vessel. The ETL pipeline correctly populated the 5 normalized tables (Ship, GT, Compressor, Propeller, High Pressure), preserving the logical decomposition designed in the approach phase.

To validate the reliability of the implemented architecture, we performed a series of integrity checks on the populated database:

1. Completeness Verification: A row-count query executed on the master table confirmed the presence of exactly 11,934 records, matching the original CSV source count. This certifies zero data loss during the ingestion process.
2. Schema Validation: We verified that all 18 features were correctly mapped to FLOAT data types, ensuring numerical precision for the subsequent decay analysis.
3. Relational Consistency: The generated primary keys (id) were correctly propagated across all sub-tables, enabling consistent JOIN operations between the ship's global parameters and the specific component metrics.

4.4 Decay Analysis Results

The execution of the temporal analysis script produced the trends visualized in Fig. 5. The visual inspection allows us to draw critical conclusions about the nature of the dataset. Our intuition was that the data is not random but follows a specific logical order. We hypothesized that the simulator generates data in a "step-wise" manner, grouping experiments by severity of wear. We used a stacked line plot to verify if this "staircase" pattern exists and how the Turbine behaves relative to the Compressor.

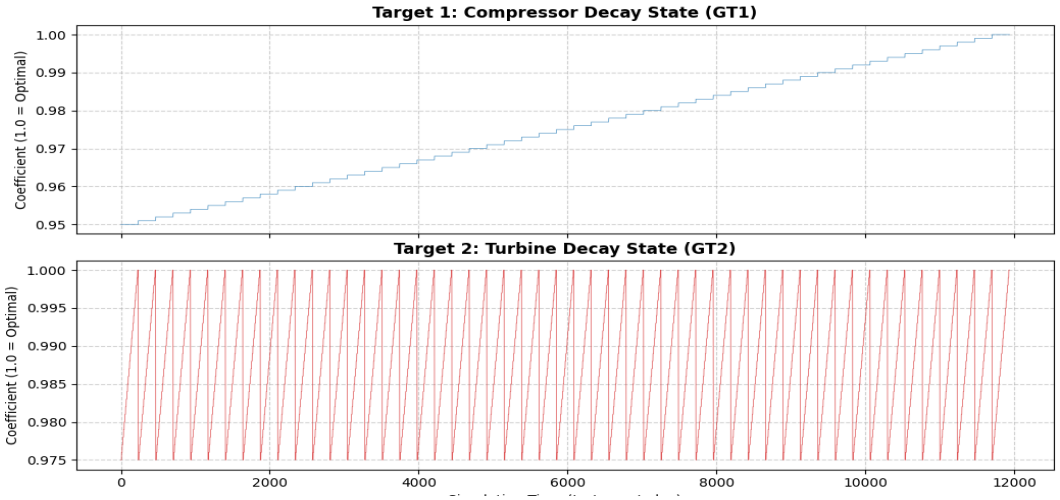


Fig. 5: Temporal Decay Analysis - Compressor (GT1) and Turbine (GT2) Coefficients.

- The Turbine Behavior: The top plot (Compressor) clearly validates our hypothesis. The decay does not happen continuously but moves in distinct steps (scales). The simulator holds the Compressor wear constant (e.g., at 0.95) for many iterations before jumping to the next level. This confirms the data is sorted by damage severity.
- The Turbine Behavior: The Turbine coefficient (bottom plot) fluctuates at a higher frequency within these steps, covering the range [0.975, 1.0]. This suggests that for every "step" of Compressor wear, the simulator tests the Turbine across its entire possible range of health, creating a complete grid of combinations.

4.5 Predictive Modeling Results

The analysis produced two sets of results: one for the Linear Regression baseline and one for the Random Forest comparison.

[A] Linear Regression Results - The scatter plots in Figure 7 show the results: on the x-axes we have real values (y_test) and on the y-axes we have predicted values (y3_pred).

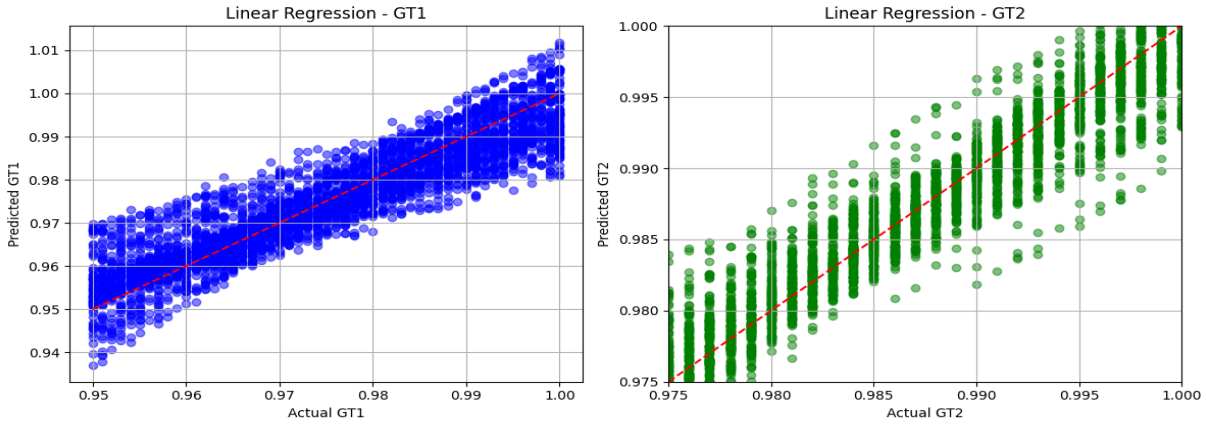


Fig. 6: Linear Regression Predictions for Compressor (left) and Turbine (right).

These scatter plots show that the Linear Regression Model fits the data almost perfectly. This indicates a strong linear relationship between the decay of the compressor/turbine and the other columns. We can confirm this by observing that the points are densely accumulated on the diagonal of the figure; very few points are slightly away from it.

The numerical results confirm this precision:

- Compressor (GT1): The Mean Absolute Error is 0.0033385 and the Mean Squared Error is 2.1715e-05.
- Turbine (GT2): The Mean Absolute Error is 0.0011895 and the Mean Squared Error is 3.2638e-06.

These numbers indicate that the error is minor for both indicators, suggesting that the Linear Regression model is highly effective for this type of data.

[B] Random Forest Comparison - Following the baseline analysis, we trained a Random Forest Regressor to further improve prediction accuracy and capture any non-linear behavior in the simulation data. The results are visualized in Figure [X], where the actual decay values are plotted against the predicted ones.

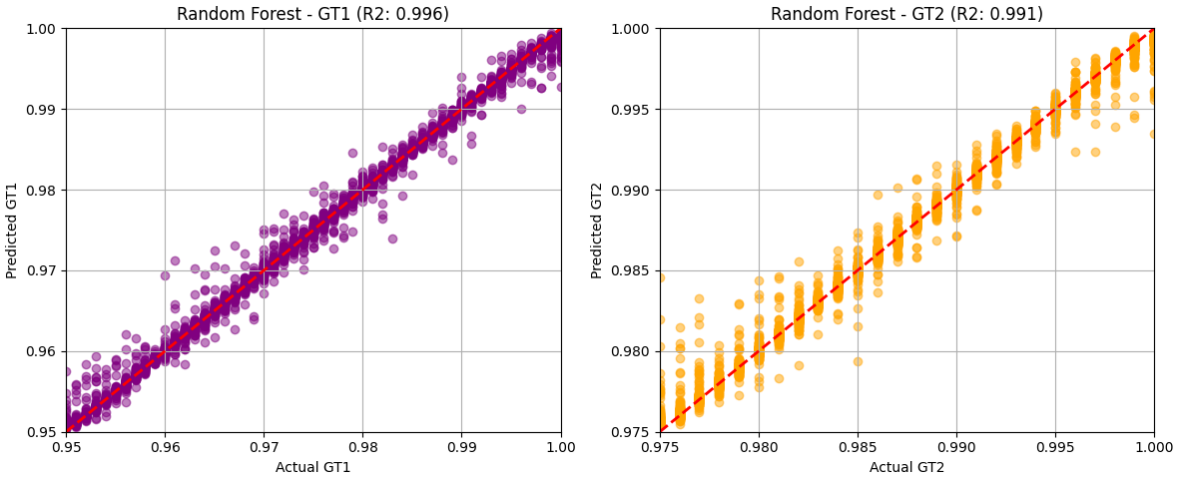


Fig. 7: Random Forest Regression Results

The purple plot (left) represents the Compressor (GT1), while the orange plot (right) represents the Turbine (GT2). As clearly shown in the figures, the predictions align almost perfectly with the red dashed diagonal line, which represents the ideal "Perfect Fit". Unlike the Linear Regression, where some dispersion was visible, the Random Forest predictions are tightly clustered, achieving an R^2 score of approximately 0.99. This indicates that the model is able to determine the exact health state of both components with negligible error, successfully handling the specific operating ranges. Consequently, the Random Forest proves to be the superior model for this predictive maintenance task.

References

[1] UCI Machine Learning Repository, "Condition Based Maintenance of Naval Propulsion Plants - https://archive.ics.uci.edu/dataset/316/condition+based+maintenance+of+naval+propulsion+plants

[2] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figueroa, "Machine Learning for Condition-Based Maintenance of Naval Propulsion Plants," *Ocean Engineering*, vol. 111, pp. 44-55, 2016.

[3] GE Aerospace, "LM2500 Marine Gas Turbine Datasheet". Available: https://www.geaerospace.com/military-defense/engines/lm2500

[4] RENK Group, "Naval Propulsion Systems - CODLAG". Available: https://www.renk.com

[5] T. Cheon, J. Kim, and S. Lee, "Impact of Sensor Faults in Hybrid Electric Propulsion Systems," *Entropy*, vol. 24, no. 12, p. 1729, 2022. Available: https://www.mdpi.com/1099-4300/24/12/1729