Fault Detection of Industrial Gas Turbine Engines

Garey Salinas
garey.salinas@colorado.edu
University of Colorado-Boulder
Boulder, Colorado, USA

Abstract

This project develops a fault detection system for industrial gas turbine engines using real-world sensor data from Kaggle. A robust preprocessing pipeline was built to engineer features (e.g., pressure ratios), normalize inputs, reduce dimensionality via PCA, and identify outliers using Isolation Forest and DBSCAN. The fault detection project employs k-fold cross-validation to train and assess a suite of classification algorithms, including Logistic Regression, Random Forest, XGBoost, and Support Vector Machines. Model explainability uses SHAP (SHapley Additive exPlanations) to identify features driving early fault onset. The proposed framework supports predictive maintenance in midstream pipeline operations by improving early failure detection, reducing downtime, and enabling targeted interventions.

Keywords

fault detection, gas turbines, feature scaling, feature engineering, dimensionality reduction, Principal Component Analysis, machine learning, Support Vector Machine, Random Forest Classifier, Logistic Regression, XGBoost, Cross Validation

1 Introduction

Industrial gas turbine engines are crucial in midstream pipeline operations, powering gas compressors. Unplanned failures can lead to significant operational disruptions, economic losses, and safety hazards. This project employs a systematic classification approach to fault detection using static sensor data. While time-series analysis was initially considered, the available dataset led to a focus on supervised learning with labeled fault conditions. Four distinct classifiers, both in their default and hyperparameter-tuned configurations, are evaluated for their effectiveness in identifying turbine faults. The methodology incorporates SHAP to determine feature importance, providing interpretable insights into the factors driving fault conditions. This data-driven diagnostic framework aims to enhance equipment reliability while potentially reducing maintenance costs and operational downtime by providing both accurate fault classifications and insights into the specific features that most significantly influence equipment health.

2 Problem Statement

While many machine learning approaches effectively classify known failure types, relatively few focus on identifying early-stage degradation that precedes faults. This is particularly true in scenarios where labeled failure data is limited. In industrial settings, vast amounts of operational sensor data are collected, but most are unlabeled. This creates an opportunity to develop models that leverage labeled and unlabeled data to detect early warning signs.

There is a clear need for scalable, interpretable fault detection systems that combine supervised classification with unsupervised anomaly detection. Such systems must produce reliable predictions while offering transparency into which features drive those predictions. This project aims to support proactive maintenance decision-making, minimize unexpected downtime, and improve operational efficiency in gas turbine-based pipeline systems by building a hybrid and explainable framework.

3 Related Work

Research by Lei et al. (2020) highlights the application of sensor data for predictive maintenance and fault detection in the industrial sector. Machine learning algorithms are extensively utilized in rotating machinery diagnostics, including Support Vector Machines (SVM), Random Forests (RF), and various deep learning models. Even rudimentary time-series data from sensors can yield favorable results.

Babu et al. (2016) emphasize the significance of feature engineering, particularly the extraction of time-domain and frequency-domain features from temperature and vibration data for effective condition monitoring. Parameters such as pressure ratios and temperature differentials have proven helpful for early failure detection.

Time-series segmentation is an evolving field of study. Malhotra et al. (2016) employed Long Short-Term Memory (LSTM) networks to identify temporal patterns, while more straightforward statistical techniques like rolling averages and trend decomposition also provide valuable insights for analyzing fault trends.

Outlier detection is a critical aspect of predictive maintenance. Techniques such as Isolation Forest (Liu et al., 2008) and DBSCAN are commonly employed to identify anomalies that could indicate impending failures.

Isolation Forest (Liu et al., 2008) has been particularly effective in detecting anomalies in high-dimensional datasets, including sensor data from industrial systems. This method isolates anomalies instead of profiling normal points, making it ideal for predictive maintenance applications. Similarly, density-based clustering methods such as DBSCAN have been widely used to identify collective outliers that could signal equipment malfunctions.

Recent industry studies have emphasized the need for real-time fault detection. Applications of Support Vector Machines (SVMs), Random Forests, and XGBoost in classifying operational data from rotating machinery have proven effective in capturing early warning signs of failure. Studies such as Ziyao et al. (2020) have made curated datasets available for developing and benchmarking fault detection models, contributing significantly to the practical advancements in this field.

Additionally, advancements in time-series analysis methods, including Long Short-Term Memory (LSTM) networks and time-series segmentation techniques, have opened new frontiers in modeling

the temporal evolution of faults. While deep learning techniques are gaining traction, classical machine learning models continue to offer robust and interpretable solutions for many industrial predictive maintenance applications.

Recent industry insights from IoT Analytics [4] emphasize the growing demand for interpretable and scalable predictive maintenance solutions. Similarly, a comprehensive review published in Applied Sciences [1] surveys machine learning techniques used in gas turbine diagnostics, reinforcing the applicability of the methods chosen in this project.

4 Proposed Work

This project proposes developing and evaluating supervised machine learning models to classify faults in industrial gas turbine engines. The selected models are Logistic Regression, Random Forest, XGBoost, and Support Vector Machines (SVM). These models are well-suited for structured sensor data and strike a balance between interpretability and predictive performance.

We will apply feature normalization using the StandardScaler to ensure consistent input scaling across features. Dimensionality reduction will be performed using Principal Component Analysis (PCA) to remove noise, reduce feature redundancy, and facilitate visualization of class separability. We aim to retain principal components that preserve at least 95%

To address the presence of anomalous data, we will use Isolation Forest and DBSCAN for outlier detection. Rather than removing outliers entirely, we will encode outlier status as new features, allowing models to account for atypical behavior during training and improve robustness.

We will apply SHAP (Shapley Additive explanations) to tree-based and kernel-based models to enhance their interpretability. This will help identify sensor-derived and engineered features, like pressure ratios and temperature ratios. These are some of the most indicative of early fault conditions, providing actionable insights for domain experts.

The dataset used in this project is the Kaggle Gas Turbine Engine Fault Detection Dataset [3], which contains labeled sensor readings (e.g., temperature, pressure, RPM) collected from gas turbine engines. Its combination of real-world signals and labeled fault conditions makes it ideal for training and evaluating supervised learning models.

The goal is to build a robust classification framework capable of detecting gas compressor failures. This is particularly relevant for midstream pipeline operations, where turbines serve as the primary driving mechanism for compression systems.

Dataset

- **Source:** Kaggle Gas Turbine Engine Fault Detection Dataset
- https://www.kaggle.com/datasets/ziya07/gas-turbine-engine-fault-detection-dataset

Data Preprocessing

- Missing value handling
- Feature Scaling using StandardScaler
- Feature engineering (pressure, speed, and temperature ratios etc.)

Dimensionality Reduction

- Principal Component Analysis

Outlier Detection

- DBSCAN
- Isolation Forest

Validation

- K-Fold Cross-Validation

Modeling Techniques

- Logistic Regression
- Random Forest Classifier
- XGBoost
- Support Vector Machines

Evaluation

- Accuracy
- Precision
- Recall
- F-1 Score

Interpretability

- Use SHAP values or feature importances to explain and interpret model predictions

Tools & Libraries

- Python (Numpy, Pandas, Scikit-learn, XGBoost, Seaborn, SHAP)
- Jupyter Notebook
- Overleaf

5 Dataset Overview

The dataset used includes 1,386 rows and 10 continuous variables, sourced from the Kaggle Gas Turbine Engine Fault Detection Dataset [3]. It captures operational parameters such as temperature, torque, pressure, vibration, RPM, power output, and fault indicators. Although fault values were originally provided as continuous measurements representing malfunction, they were converted to binary (boolean) indicators for this classification task, where 0 represents normal operation and 1 indicates fault conditions. All entries are complete and well-suited for classification modeling.

6 Dataset Characteristics and Exploratory Analysis

The dataset comprises 1,386 observations and 10 features capturing turbine operation metrics, including:

- **Temperature, RPM, Torque, Power Output** – Continuous variables describing mechanical conditions.
- **Vibrations, Fuel Flow Rate, Air Pressure, Exhaust Gas Temperature, Oil Temperature** – Sensor readings essential for fault diagnostics.
- **Fault (target)** – Binary label where 0 indicates normal operation and 1 indicates fault.

All columns are complete with no missing values. The target variable is imbalanced: 69% normal vs. 31% fault cases. The class imbalance in the `Fault` variable was visualized using a bar plot. Out of 1,386 samples, 31% represent faulty conditions. This imbalance motivated the use of stratified k-fold cross-validation during model evaluation to maintain representative distributions across folds.

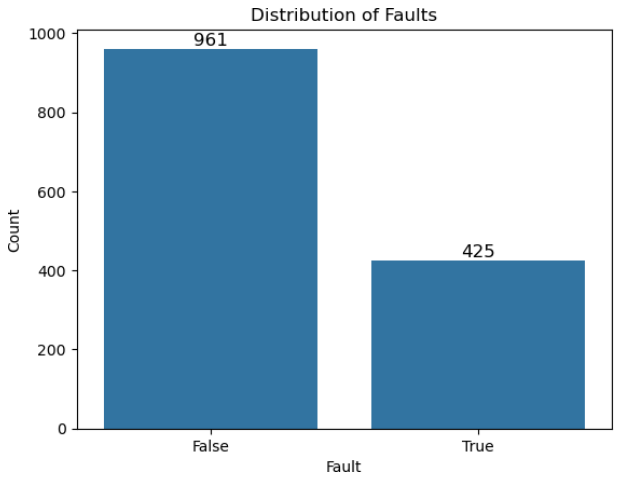

Figure 1: Distribution of Fault Labels (0 = Normal, 1 = Fault)

7 Statistical Summary

- **Temperature**: Mean ≈ 902°C, with most values centered around this point.
- **RPM**: Tightly clustered around 15,000 RPM (Mean: 15,022.6; Std: 490.1).
- **Torque**: Majority between 3,355 and 3,627 Nm, centered around 3,495 Nm.
- **Vibrations**: Slight right skew; most readings between 1.6 and 2.3 mm/s.
- **Power Output**: Centered around 100 MW with moderate spread.
- **Fuel Flow Rate**: Mostly between 2.3 and 2.7 kg/s.
- **Air Pressure, Exhaust Gas Temperature, Oil Temperature**: All features show narrow, symmetric distributions, with values consistent with typical operational thresholds.

8 Feature Engineering

This project implemented multiple engineered domain-specific ratios to better reflect real-world mechanical relationships. New features included:

- `exhaust_efficiency_ratio`
- `rpm_torque_ratio`
- `vibration_rpm_ratio`
- `oil_temp_torque_ratio`
- `power_torque_ratio`
- `oil_temp_rpm_ratio`
- `oil_temp_power_ratio`
- `oil_temp_vibrations_ratio`
- `oil_temp_exhaust_ratio`
- `power_vibrations_ratio`
- `torque_fuel_ratio`
- `comp_pressure_ratio`
- `fuel_flow_per_rpm`
- `vibrations_torque_ratio`
- `temperature_torque_ratio`
- `exhaust_temp_torque_ratio`
- `air_pressure_torque_ratio`

These helped expose subtle patterns associated with early degradation. The histogram visualization contains a series of histograms for all numeric features, including both original sensor readings and engineered features. Each plot visualizes the distribution of values for a specific feature, highlighting their central tendency, spread, and shape. For a full overview of the distributions of the scaled features after engineering, refer to the histograms provided in Appendix 2. Key Observations include:

- **Original Features (e.g., Temperature, RPM, Torque):** Most original features, such as temperature, rpm, and torque, exhibit approximately normal distributions centered around their respective means, with slight variations in spread. Features like vibrations show a slight right skew, indicating the presence of higher values in the tail.
- **Engineered Features (e.g., Ratios):** Many engineered features, such as `power_torque_ratio` and `fuel_flow_per_rpm`, show symmetric distributions, while some exhibit narrower or wider spreads compared to the original features. Features like `oil_temp_vibrations_ratio` display right-skewed distributions, indicating the presence of extreme values or outliers.
- **Skewness and Outliers:** A few features, such as `oil_temp_vibrations_ratio` and `fuel_flow_per_rpm`, display noticeable skewness with long tails, suggesting operational anomalies or outliers.
- **Feature Variability:** Distributions vary in range and density, reflecting the diversity of the dataset's features. Features like `air_pressure` and `rpm` are tightly clustered around their means, whereas ratios such as `vibrations_torque_ratio` show smaller ranges.
- **Insights for Fault Detection:** Despite some variability, most feature distributions overlap significantly, making it challenging to distinguish faults based on individual features alone. The engineered ratios, however, provide potential additional insights into operational dynamics and early fault detection.

9 Correlation

Understanding the relationships between different sensor measurements and engineered features is critical for effective feature selection and fault detection modeling. A Pearson correlation heatmap was generated to visually assess the linear dependencies among variables. Key observations include:

- **Strong positive correlation** between RPM and Torque, indicating engine load increases proportionally with rotational speed.

- **High correlation** between Fuel Flow Rate and Power Output, suggesting energy generation is closely linked to fuel consumption.
- **Moderate negative correlation** between Exhaust Temperature and Efficiency Ratio, implying that higher temperatures may reduce operational efficiency.
- **Lower correlations** for newly engineered features, suggesting the capture of novel operational behaviors.

These correlation insights were leveraged during model feature selection to prioritize variables contributing the most to accurate fault classification. A full-sized version of the correlation heatmap is provided in Appendix 3.

10 Dimensionality Reduction

Principal Component Analysis (PCA) was applied to the dataset to reduce feature dimensionality while preserving variance. This transformation aimed to enhance model efficiency, mitigate multicollinearity, and facilitate visualization. Key observations from the PCA results include:

- **Steep Climb (PC1 to PC6):**
  The first six principal components explain a substantial portion of the variance, with each contributing significant new information.
- **Elbow Point (Component 6 or 7):**
  After the sixth component, additional components contribute less incremental value. The "elbow," where the gain in explained variance starts to level off, occurs around Principal Component 6 to 7.
- **95% Variance Threshold at PC8:**
  The cumulative explained variance curve intersects the 95% threshold at the eighth principal component. Thus, the first eight components together retain at least 95% of the total variance in the data.

Based on these observations, dimensionality reduction using PCA effectively preserved the majority of the original dataset's informational content with a reduced set of features. A visual summary of the explained variance ratios and cumulative variance captured by each principal component is provided in Appendix 4.

11 Outlier Detection

Outliers were identified using both Isolation Forest and DBSCAN. Rather than excluding these anomalies, two binary features were created to reflect their detected presence, enriching the learning process. For DBSCAN, a k-distance graph was generated by plotting the distance to each point's 5th nearest neighbor, sorted in ascending order. Upon analyzing the k-distance graph, the sharpest increase in distance, known as the "elbow," occurs at approximately $\varepsilon \approx 4.7$. This represents the ideal threshold for DBSCAN. A visual depiction of the k-distance elbow plot is provided in Appendix 5.

To further illustrate the effectiveness of outlier detection, a two-panel scatter plot was generated comparing Isolation Forest and DBSCAN results. In the plots, normal points are shown in blue, while outliers are highlighted in red. Isolation Forest identifies outliers more broadly distributed across the feature space, with several extreme points marked in the outer regions, whereas DBSCAN identifies a smaller, more concentrated set of outliers, focusing on points significantly separated from dense clusters. A visualization comparing the two methods' detected outliers is provided in Appendix 6.

12 Model Training and Evaluation

Four classification models were trained and evaluated using 5-fold cross-validation: Logistic Regression, Random Forest, XGBoost, and Support Vector Machines (SVM). Key metrics assessed included accuracy, precision, recall, and F1-score.

The model evaluation results show relatively modest changes between original and tuned models across all metrics. SVM achieved the highest accuracy (0.6926) in its original form, while Logistic Regression maintained nearly identical performance before and after tuning (0.6912). Although Random Forest demonstrated the best precision (0.5915) in its original configuration, the tuned SVM showed the most balanced performance overall with the highest F1 Score (0.5940) after tuning. Interestingly, XGBoost performed well in its original form with the highest original F1 Score (0.5948), but showed slight decreases in performance after hyperparameter tuning. These results suggest that while hyperparameter tuning provided some benefits, particularly for SVM's balance between precision and recall, the improvements were incremental rather than transformative across all models. Among these, the tuned SVM emerged as the most balanced model with the highest F1-score, while the original XGBoost model offered stable and interpretable performance. A visualization comparing the model metrics is provided in Appendix 7.

13 Model Explainability with SHAP

The SHAP analysis reveals a clear hierarchy of feature importance in our engine fault detection model. The `oil_temp_exhaust_ratio` emerged as the most influential predictor, demonstrating that the relationship between oil temperature and exhaust temperature provides critical diagnostic information. This suggests that imbalances in these thermal components are strongly indicative of potential engine malfunctions.

Mechanical load indicators, particularly `torque`, ranked second in importance, confirming its direct relationship with engine performance outcomes. `temperature` readings independently ranked third, underscoring the fundamental role of thermal monitoring in predictive maintenance. `power_output` and `vibrations_rpm_ratio` rounded out the top five features, highlighting how operational performance metrics and stability indicators contribute substantially to fault prediction.

The model also identified several ratio-based features as moderately important, including `exhaust_temp_torque_ratio` and `air_pressure_torque_ratio`, which combine thermal and pressure measurements with mechanical load data. Speed-related parameters (rpm and its derived ratios) demonstrated meaningful but comparatively lower predictive power. This comprehensive feature importance hierarchy provides valuable insights for focusing condition monitoring efforts on the most diagnostically relevant engine parameters. These insights provide actionable feedback to engineers for early interventions and fault diagnosis. A visualization of a SHAP dot is provided in Appendix 8.

14 Timeline

This timeline outlines the key phases of the project, beginning with literature review and data preparation, followed by model development and anomaly detection. The final days are allocated to compiling results and preparing the final report and presentation. Table 1 summarizes the major project milestones. The project is scheduled for completion by April 28, 2025, aligning with the course submission deadline.

15 Discussion

The objective of this project was initially to introduce time series analysis for fault detection. The goal was to use the temporal patterns in sensor data to better understand equipment behavior prior to failures. After a search, finding a good time-series labeled dataset was challenging. The result was that the project used snapshot sensor data to derive a model.

Nevertheless, the value of online time series data for predictive maintenance is significant. Including "Real-time" sensor streams in the models would allow them to adapt to dynamic changes in equipment deterioration. This would enable the detection of anomalies on the fly instead of requiring static data.

Real-time fault detection might significantly improve operational decision-making. Maintenance could be scheduled more efficiently and with less impact on system availability, which would reduce overall system reliability and utilization. Future research in this field could use real-time monitoring and a learning model that is adaptive to new data. Streaming-based data analytics could mean the emergence of early prediction systems that identify issues and suggest the most effective compliant mobilization schedules based on prevailing operating conditions.

Further research could include time-series information that provides for operational cycles, environmental conditions (ambient temperature, humidity, and levels of induced vibration), and data inputs from an operator over time. Recent methods are RNNs, LSTMs, and Transformer-based models for time-series prediction. Applications could head toward multi-dimensional time-series classification, sequence anomaly detection, and predictive maintenance guided by sequence modeling. To scale the experiments, they must be implemented on simulated or real turbine sensor streams in a controlled environment and validated in real-world contexts and across various engines and operational conditions.

Several promising directions for extending this research include implementing SHAP for feature explainability and model transparency across all models, analyzing misclassification patterns through targeted error analysis to identify specific operational conditions where models struggle, exploring ensemble models (e.g., stacking or voting classifiers) to leverage strengths of individual algorithms, evaluating models in a simulated streaming context for real-time fault detection capabilities, and extending the dataset with time-based features or additional contextual variables to capture more complex operational patterns.

In general, even though this project's scope was constrained to existing data, the experience pointed out the increasing relevance of time-series data for developing predictive maintenance for industrial use.

16 Conclusion

Of all the models tested, the SVM (Tuned) emerged as the most balanced algorithm, demonstrating the highest F1 Score and competitive precision and recall metrics. Original XGBoost also delivered consistent and interpretable performance, solidifying its role as a strong and trustworthy baseline for fault detection. In general, tuning enhanced model balance, particularly for SVM, although not all models benefited equally from hyperparameter optimization. This highlights the need for both effective model selection and context-aware tuning strategies when implementing fault detection systems.

In addition to model evaluation, incorporating SHAP-based feature importance analysis added a critical layer of explainability. This allowed for global insights into which features most strongly influenced model behavior, with `oil_temp_exhaust_ratio`, `torque`, and `temperature` emerging as the leading predictors across all models. The analysis validated our domain-engineered features, with ratios like `air_pressure_torque_ratio` and `rpm_torque_ratio` ranking among the top predictive features despite being created through feature engineering, proving their relevance in real-world operational diagnostics. At the local level, force plots and dependence plots provided individualized explanations—revealing how low `fuel_flow_rate` and `torque` in certain instances signaled higher fault likelihoods, while high `rpm` and thermal efficiency metrics worked as counterbalancing factors.

Overall, this project demonstrates that fault detection in gas turbines can benefit significantly from combining supervised learning with model explainability, yielding not just predictions but actionable insights to guide maintenance, reduce downtime, and enhance safety in industrial environments. We believe that future work with real-time data and adaptive models can further increase such systems' predictive power and operational significance.

17 Citations and Bibliographies

This project builds on a range of research studies and industry data sources. The foundational background includes a comprehensive survey on anomaly detection by Chandola et al. [2] and a systematic review of machinery health prognostics by Lei et al. [5]. Isolation Forest, introduced by Liu et al. [6], is one of the primary algorithms utilized in this project for unsupervised anomaly detection.

Recent market insights from IoT Analytics [4] underscore the increasing demand for predictive maintenance systems across industrial applications. Additionally, a review published in *Applied Sciences* [1] provides a current overview of machine learning models specifically applied to gas turbine anomaly detection, reinforcing the relevance of the selected techniques.

The dataset used for experimentation is the Gas Turbine Engine Fault Detection Dataset hosted on Kaggle [3], which provides real-world sensor data and labeled fault types suitable for supervised classification tasks.

References

[1] Various Authors. 2024. Review of Machine Learning Models for Gas Turbine Anomaly Detection. *Applied Sciences* 14, 11 (2024), 4551. doi:10.3390/app14114551

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys (CSUR)* 41, 3 (2009), 1–58. doi:10.1145/1541880.1541882

Table 1: Proposed Project Timeline

Phase	Dates	Tasks
Project Planning & Literature Review	Apr 16 – Apr 18	Finalize proposal, conduct literature review, and outline the modeling approach
Data Preprocessing	Apr 18 – Apr 21	Clean dataset, perform exploratory data analysis (EDA), scale features, engineer new variables, apply PCA, and detect outliers
Model Building, Evaluation, Feature Importance	Apr 21 – Apr 24	Train and evaluate classification models (Logistic Regression, Random Forest, XGBoost, SVM), assess performance metrics, and analyze feature importance using SHAP
Final Report & Presentation	Apr 25 – Apr 28	Compile findings, create visualizations, write the final report, and prepare presentation slides

[3] Ziyao Chen. 2020. Gas Turbine Engine Fault Detection Dataset. https://www.kaggle.com/datasets/ziya07/gas-turbine-engine-fault-detection-dataset. Accessed: 2025-04-16.

[4] IoT Analytics. 2023. Predictive Maintenance Market Report 2023. https://iot-analytics.com/predictive-maintenance-market. Accessed: 2025-04-16.

[5] Yaguo Lei, Naipeng Li, Lin Guo, Ning Li, Tao Yan, and Jing Lin. 2020. Machinery Health Prognostics: A Systematic Review from Data Acquisition to RUL Prediction. *Mechanical Systems and Signal Processing* 138 (2020), 106761. doi:10.1016/j.ymssp.2019.106761

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422. doi:10.1109/ICDM.2008.17

A Additional Visualizations

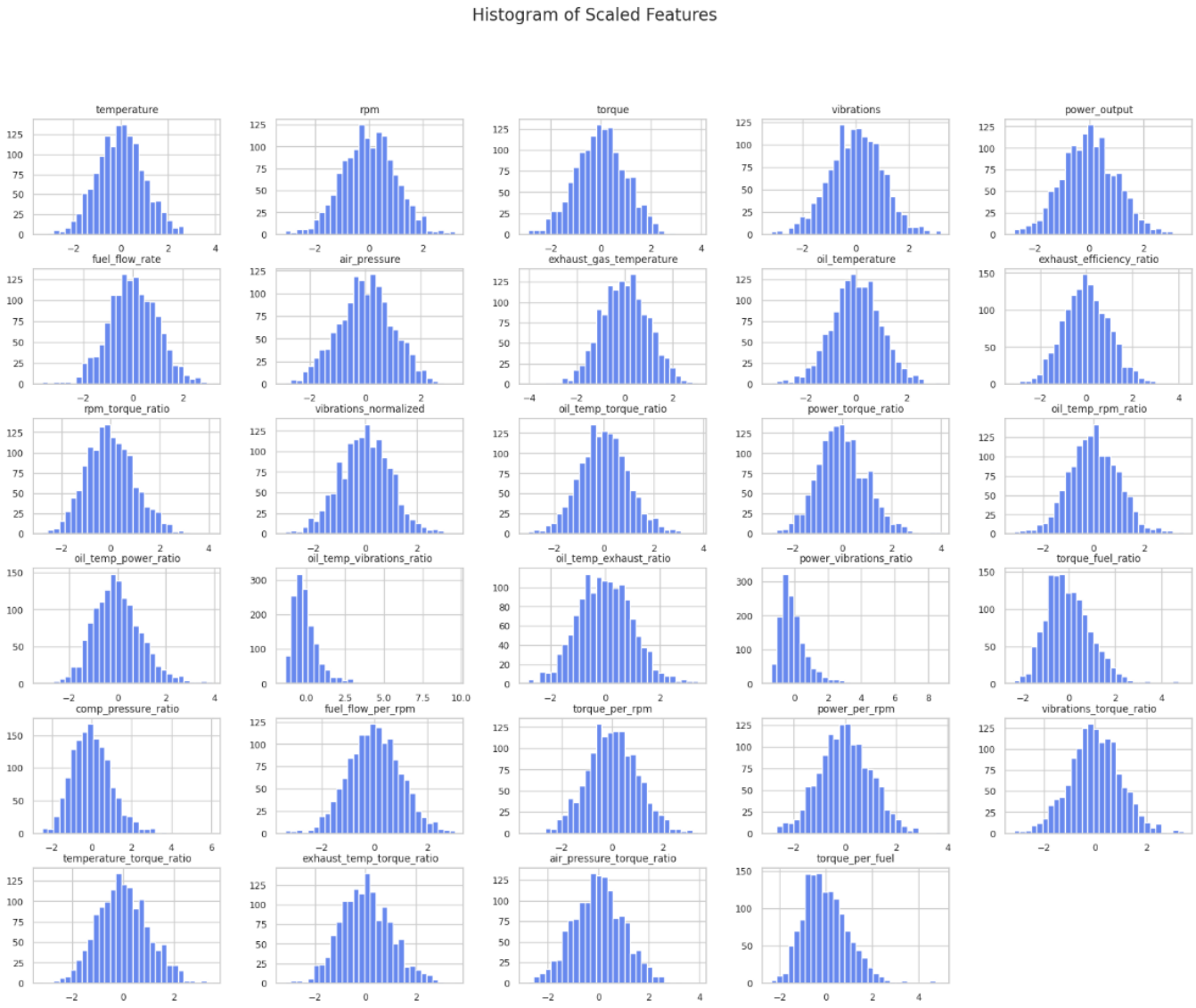

Figure 2: Histograms for all numeric features, including original sensor readings and engineered features. Each plot visualizes the distribution's central tendency, spread, and shape. Referenced in Section 8.

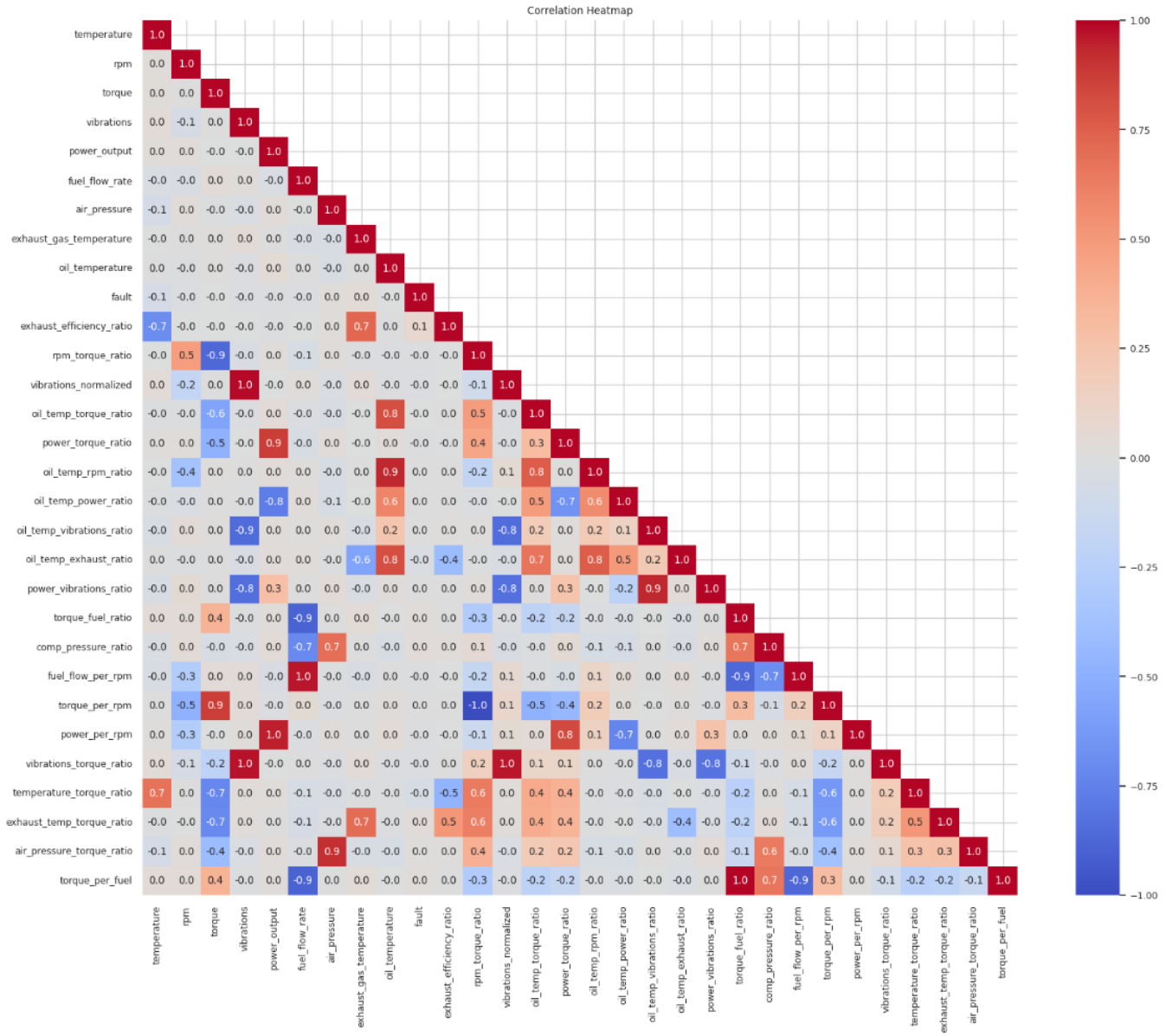

Figure 3: Correlation Heatmap of Gas Turbine Sensor Features and Engineered Variables. The visualization reveals several key relationships: strong positive correlations between thermal features (`temperature`, `exhaust_gas_temperature`) and between mechanical parameters (`power_output`, `torque`); weak correlations between most engineered features and the target fault variable, suggesting the need for multivariate modeling; moderate negative correlations between operational parameters like `fuel_flow_per_rpm` and `comp_pressure_ratio`; and distinct feature clusters that indicate potential redundancy addressable through dimensionality reduction. Color intensity represents correlation strength, with dark blue showing strong negative correlations and dark red indicating strong positive correlations. Referenced in Section 9.

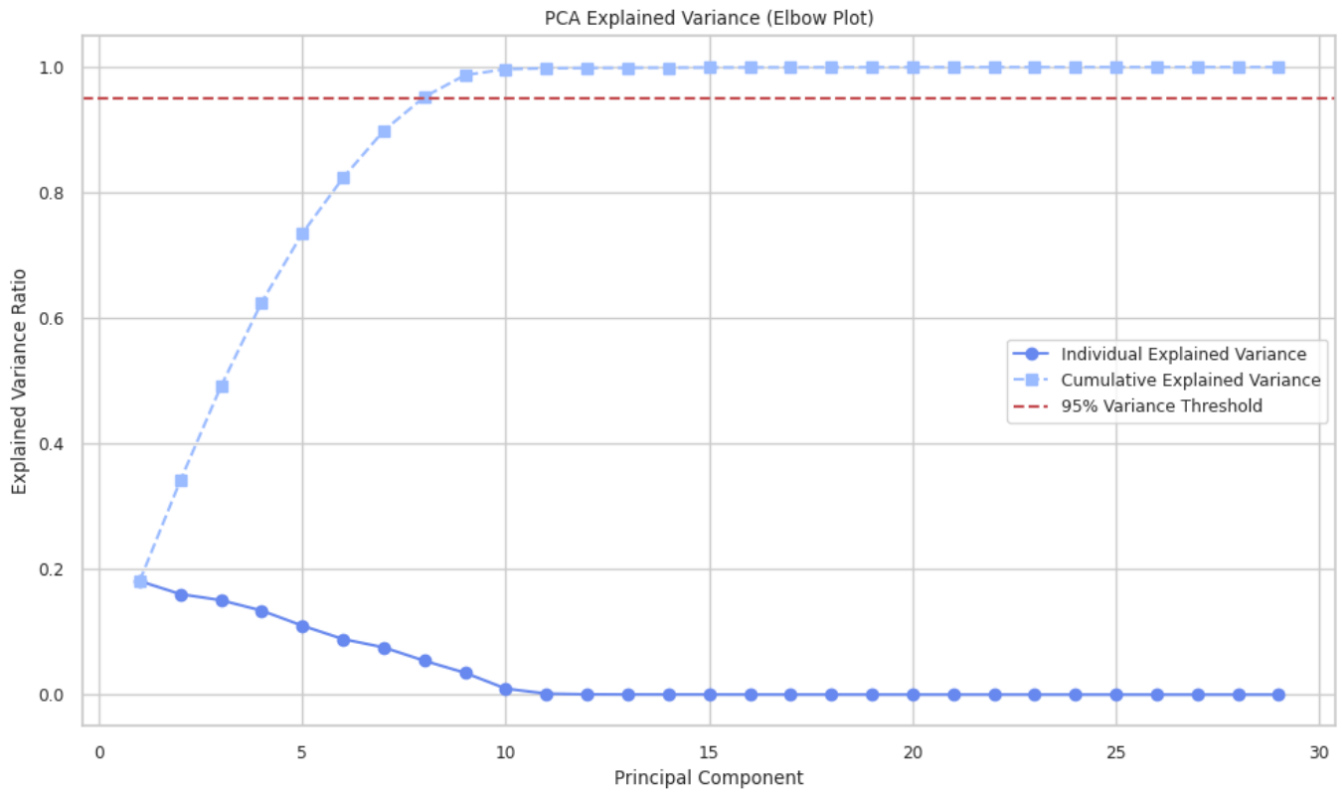

Figure 4: PCA Explained Variance Analysis with Elbow Plot. The figure displays both individual (blue bars) and cumulative (orange line) explained variance ratios across principal components. The first six components show a steep climb in explained variance, with each contributing significant information. An "elbow" point occurs around PC6-7, where the incremental gain in explained variance begins to level off. The red dashed line indicates the 95% variance threshold, which is reached at the eighth principal component. This analysis supports dimensionality reduction to eight principal components while preserving most of the dataset's informational content. Referenced in Section 10.

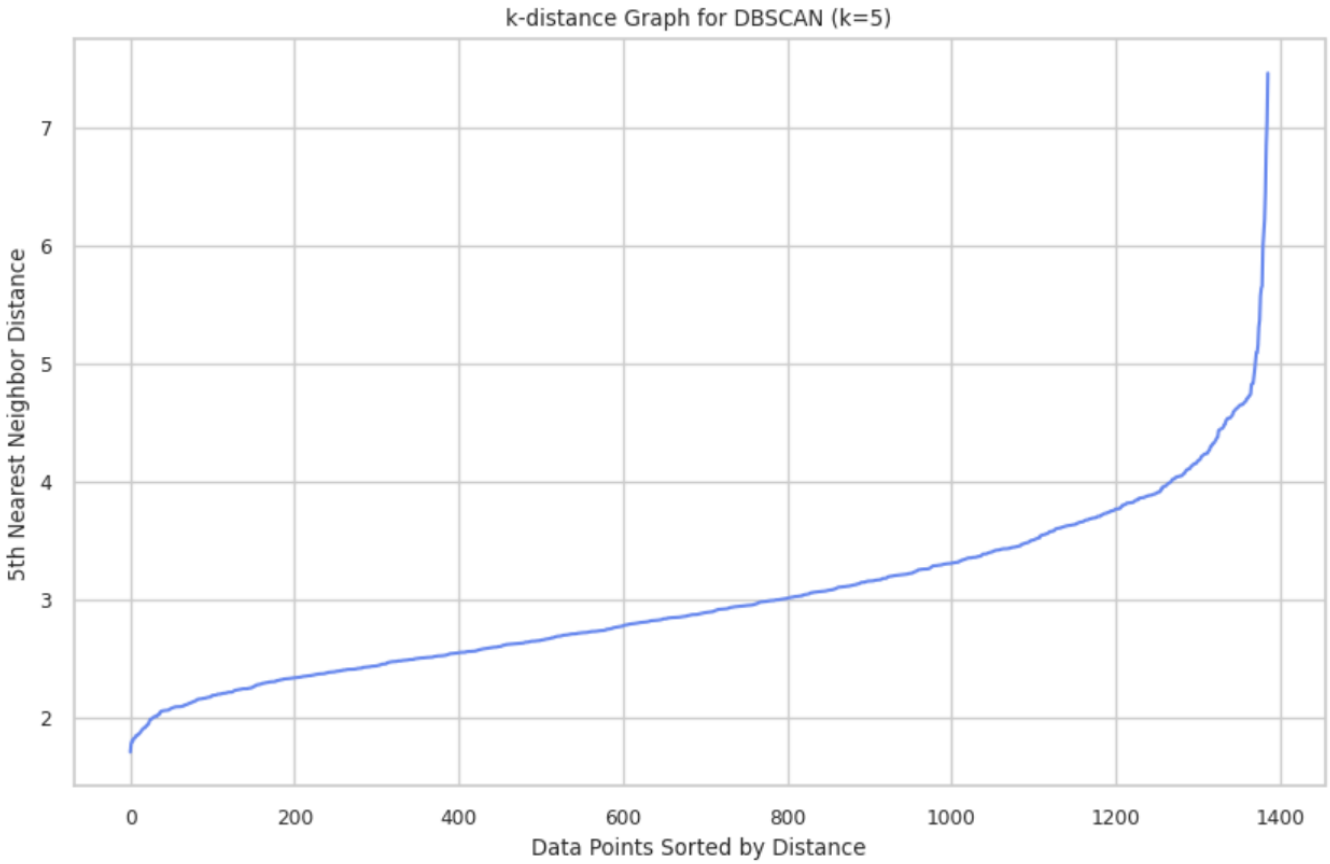

Figure 5: DBSCAN k-distance Graph for Outlier Detection Parameter Selection. The plot shows distances to each point's 5th nearest neighbor, sorted in ascending order. The "elbow" point, where distances begin increasing sharply, occurs at approximately $\varepsilon \approx 4.7$. This threshold represents the optimal parameter value for DBSCAN, allowing the algorithm to effectively capture dense clusters while identifying isolated points as outliers. The steep slope after this threshold indicates a natural separation between core cluster points and potential anomalies. Referenced in Section 11.

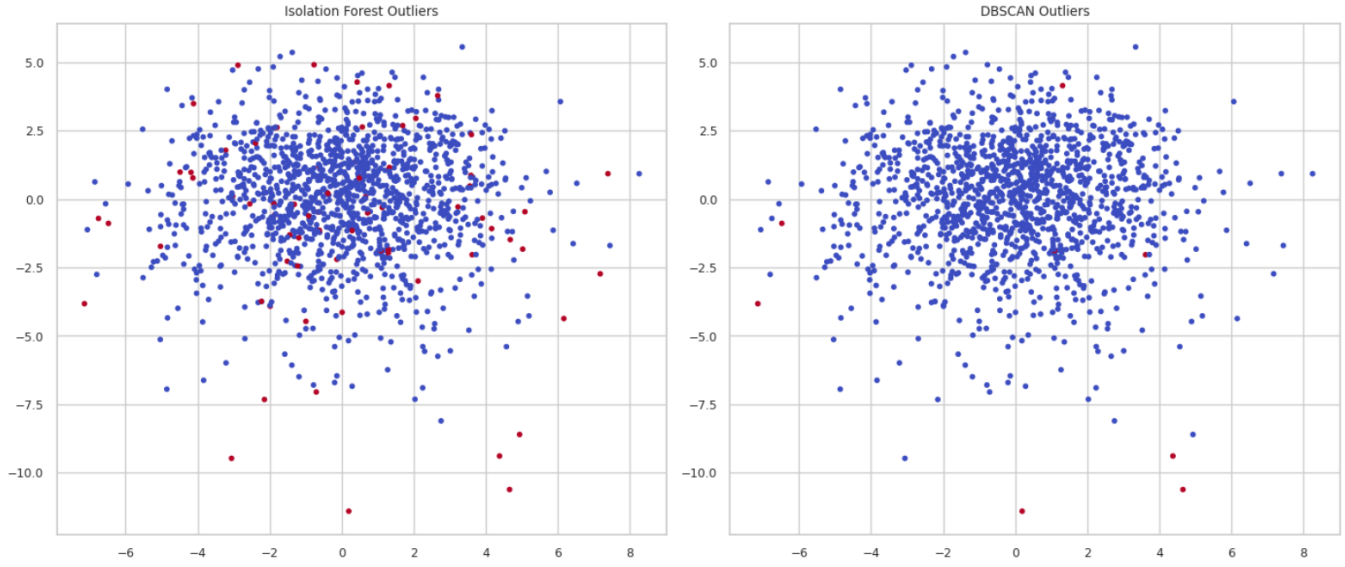

Figure 6: Comparison of Outlier Detection Results. Left: Isolation Forest. Right: DBSCAN. Normal points are shown in blue; detected outliers are highlighted in red. Referenced in Section 11.

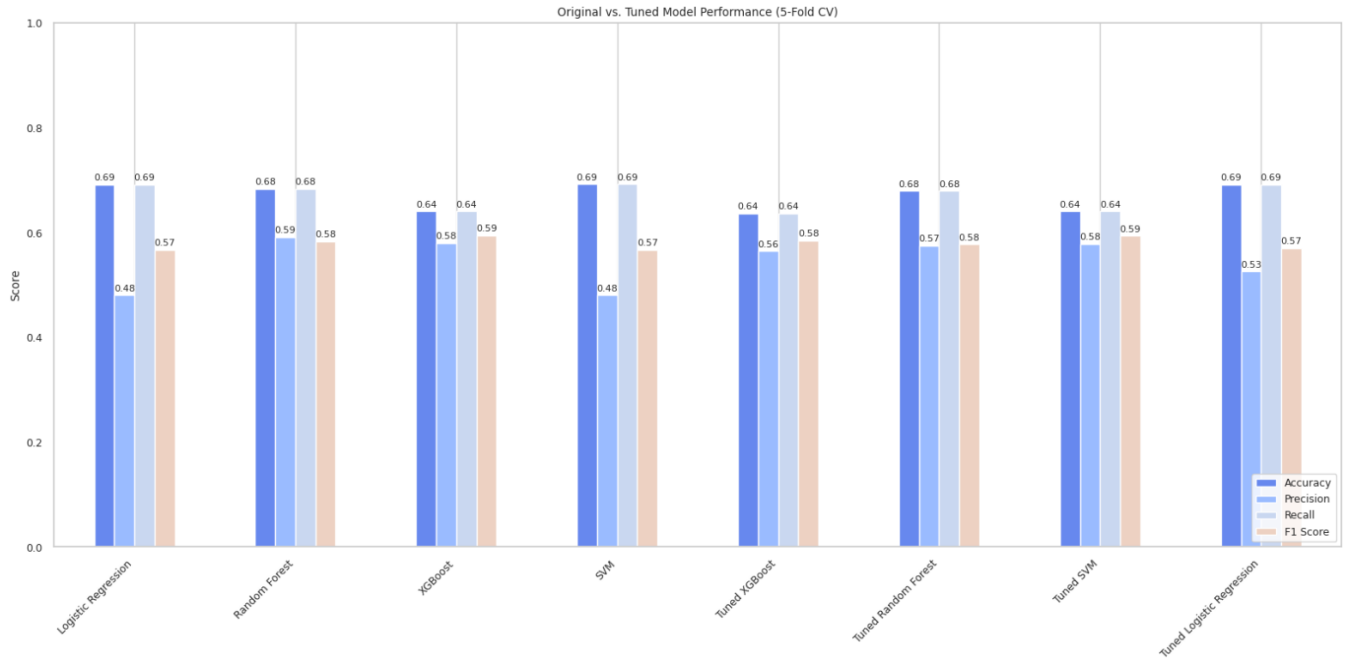

Figure 7: Comparison of Original and Tuned Model Performance Across Four Classification Models. The bar chart displays accuracy, precision, recall, and F1 score metrics for Logistic Regression, Random Forest, XGBoost, and SVM models before and after hyperparameter tuning, based on 5-fold cross-validation results. The visualization reveals modest differences between original and tuned model performance, with SVM and Logistic Regression achieving the highest accuracy scores (0.69), while precision values show greater variation across models. Referenced in Section 12.

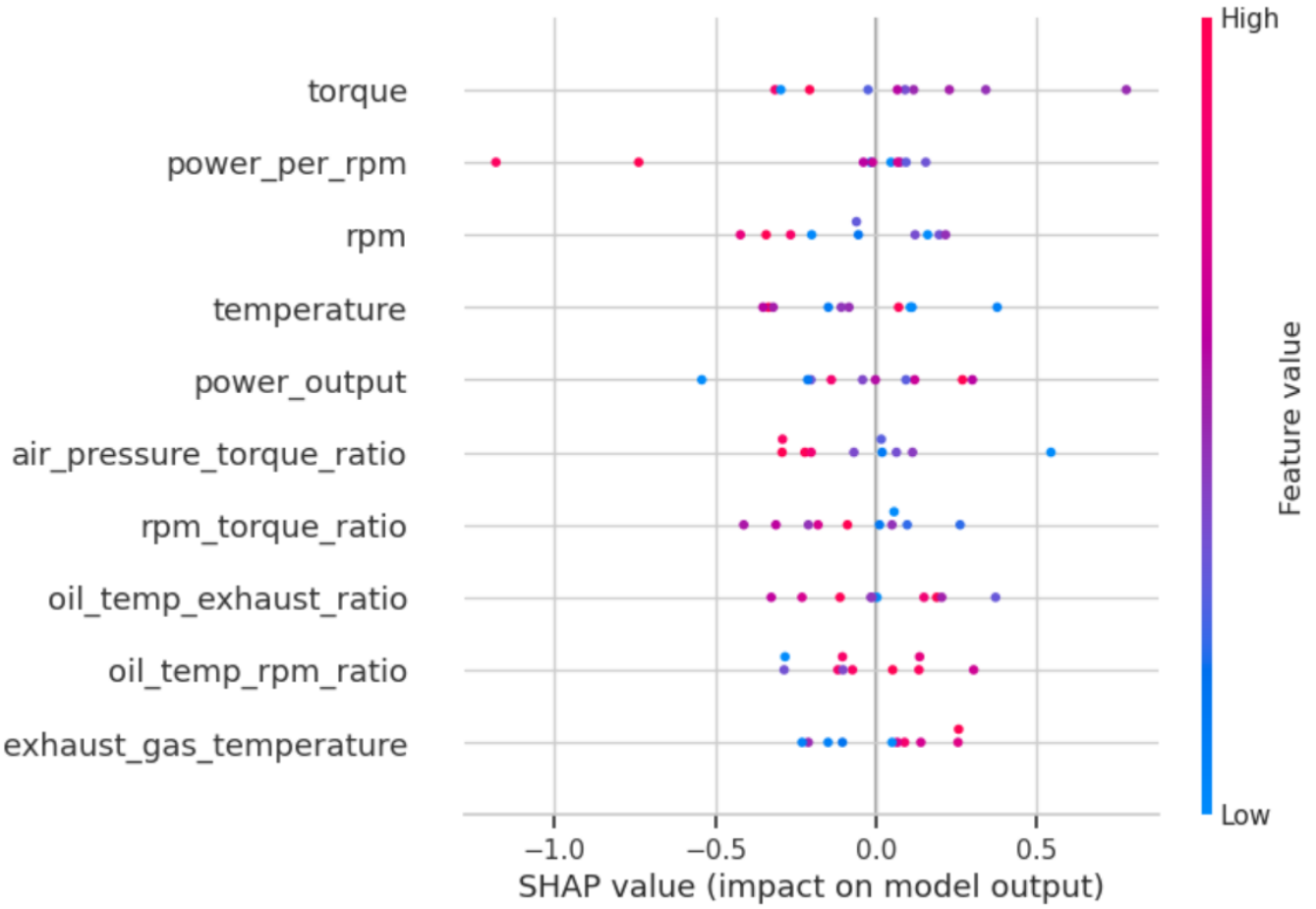

Figure 8: SHAP Summary Dot Plot Showing Feature Impact on Fault Prediction. Each point represents a single instance in the dataset, with horizontal position indicating the SHAP value (impact on model prediction) and color representing the feature value (pink/red = high, blue = low). Features are ranked by importance, with those at the top having the greatest overall influence on model output. The plot reveals that high `torque` values typically push predictions toward fault conditions, while features like `power_per_rpm` show more complex, non-linear relationships where both high and low values can influence predictions in different directions depending on context. Referenced in Section 13.