algorithms

MDPI

Article

Intelligent Multi-Fault Diagnosis for a Simplified Aircraft Fuel System

Jiajin Li *, Steve King and Ian Jennions *

Integrated Vehicle Health Management Centre, School of Aerospace, Transport, and Manufacturing, Cranfield University, Bedfordshire MK43 0AL, UK; s.p.king@cranfield.ac.uk

* Correspondence: jiajin.li@cranfield.ac.uk (J.L.); i.jennions@cranfield.ac.uk (I.J.)

Abstract: Machine learning (ML) techniques are increasingly used to diagnose faults in aerospace applications, but diagnosing multiple faults in aircraft fuel systems (AFSs) remains challenging due to complex component interactions. This paper evaluates the accuracy and introduces an innovative approach to quantify and compare the interpretability of four ML classification methods—artificial neural networks (ANNs), support vector machines (SVMs), decision trees (DTs), and logistic regressions (LRs)—for diagnosing fault combinations present in AFSs. While the ANN achieved the highest diagnostic accuracy at 90%, surpassing other methods, its interpretability was limited. By contrast, the decision tree model showed an 82% consistency between global explanations and engineering insights, highlighting its advantage in interpretability despite the lower accuracy. Interpretability was assessed using two widely accepted tools, LIME and SHAP, alongside engineering understanding. These findings underscore a trade-off between prediction accuracy and interpretability, which is critical for trust in ML applications in aerospace. Although an ANN can deliver high diagnostic accuracy, a decision tree offers more transparent results, facilitating better alignment with engineering expectations even at a slight cost to accuracy.

Keywords: aircraft fuel system; multiple fault diagnosis; machine learning; interpretability; explainable AI

check for updates

Academic Editors: Mateus Mendes, Balduíno Mateus and Nuno Lavado

Received: 28 November 2024
Revised: 16 January 2025
Accepted: 21 January 2025
Published: 1 February 2025

Citation: Li, J.; King, S.; Jennions, I. Intelligent Multi-Fault Diagnosis for a Simplified Aircraft Fuel System. *Algorithms* **2025**, *18*, 73. https://doi.org/10.3390/a18020073

1. Introduction

An aircraft fuel system (AFS) performs three primary functions: fuel storage, fuel supply to the engine, and fuel management and distribution. Fuel is stored in multiple tanks throughout the fuselage and wings, designed to withstand vibrations, pressure changes, and various in-flight loads. The AFS ensures clean fuel delivery to the main engines and the auxiliary power unit (APU), maintaining thrust under all operating conditions, whether in flight or on the ground [1]. To provide reliable fuel delivery, AFS designs often include redundancies; for instance, the Boeing 737 has two boost pumps and bypass lines per tank, allowing gravity-fed cross-feeding across engines on both sides [2]. Additionally, the AFS helps control the aircraft's centre of gravity, reducing flight resistance and fuel consumption [3], and regulates fuel temperature via the fuel oil heat exchanger (FOHE) to optimise performance.

To fulfil its functions, the aircraft fuel system (AFS) is equipped with multiple components, including fuel tanks, pumps, valves, pipelines, filters, fuel metering units, injectors, and other essential equipment. Additionally, the AFS incorporates various sensors to monitor critical parameters such as fuel level, pressure, flow rate, and temperature, providing real-time data on the system's operational status. These sensors serve as the primary data source for diagnosing faults, enabling a timely detection of issues and supporting the reliability and safety of fuel delivery.

During normal operation, components within the aircraft fuel system (AFS) undergo continuous degradation at varying rates, which can eventually lead to functional failures. Common failure modes include tank pressure failure, leakage, blockage, and partial or complete fuel pump loss, each of which impacts sensor readings differently. Detecting, isolating, and repairing faults in the AFS promptly is essential. Therefore, any fault detection system must provide robust and accurate predictions and produce interpretable results for domain operators to minimise ambiguity. Additionally, since multiple fault modes may coexist, it is critical to study AFS fault modes in the context of simultaneous component degradation or failure.

ML-based fault diagnosis has garnered significant attention in industry applications. Previous work in this study ([4]) has identified several machine learning (ML) techniques with the potential for classifying faults in complex systems like an aircraft fuel system (AFS). For a comprehensive review of ML methods applicable to fault diagnosis, readers may refer to this prior work. Choosing an effective ML algorithm requires understanding dataset size requirements, as training data quantity influences algorithm performance and overfitting risk. While traditional machine learning (ML) typically requires less data than deep learning, which involves neural networks with multiple layers of nonlinear transformations, complex ML algorithms, often considered traditional ML models with complex rules or ensemble learning approaches, still require larger datasets and more thorough feature engineering. Limited systematic research on dataset size effects exists; however, one study on medical datasets compared six supervised ML algorithms [5], finding that data distribution affected algorithm performance more than dataset size. Simpler algorithms like naive Bayes performed most robustly on smaller datasets, followed by support vector machines (SVMs) and then neural networks (NNs), while decision trees (DTs) showed less robustness but a slightly higher average accuracy than SVMs. The approximate data requirements for common ML algorithms, in ascending order, are as follows: simpler algorithms (e.g., logistic regression, naive Bayes) < decision trees < SVMs < NNs or random forests.

While existing ML-based fault diagnosis algorithms can accurately detect individual faults, system-level diagnosis in systems with multiple degradations or faults often still requires multi-algorithm methods [6] and parallel computing [7], which can be resource-intensive in terms of storage and computation. Advances in multiclass ML algorithms have enabled researchers to identify up to seven distinct failure modes [8–12] or varying severity levels of a single fault [13], though, typically, under the assumption of only one active fault in the system. More recently, studies have begun to explore diagnosing fault combinations; for instance, ref. [14] examined six failure modes and a limited set of dual-fault combinations. However, a comprehensive approach capable of diagnosing all possible fault combinations within a complex system, such as an aircraft fuel system, remains lacking. Developing a robust multi-fault diagnosis method that can efficiently handle a full range of fault combinations is essential and represents the primary challenge this paper seeks to address.

Machine learning (ML) algorithms are becoming increasingly complex to meet higher expectations for predictive accuracy. However, the opaque decision-making processes of these sophisticated algorithms often make their outputs difficult for human users to interpret. In contrast, transparent algorithms like decision trees (DTs) and logistic regressions (LRs) offer more accessible decision logic. For instance, [15] assessed the interpretability of DTs and LRs in breast cancer diagnosis using identity, stability, and separability metrics, finding that DTs had superior interpretability. Similarly, a study on Alzheimer's disease [16] ranked tree-based algorithms as the most interpretable, followed by multilayer perceptron (MLP), with support vector machines (SVMs) and k-nearest neighbours (KNNs) lagging. This suggests a preliminary interpretability ranking of DTs > LRs > MLPs > SVMs/KNNs. Limited interpretability can hinder the broader adoption of ML-based diagnostic algorithms [17,18], particularly in high-stakes fields like fault diagnosis [19]. Explainable AI (XAI) shows promise in enhancing user trust [20] and providing insights into ML algorithm learning [21], yet its use in fault diagnosis remains scarce [22]. Furthermore, as noted in [4], XAI alone may not fully address trust concerns ([23,24]) unless key limitations are resolved, specifically (1) developing robust metrics to evaluate explanations [20] and (2) validating XAI outputs. This paper seeks to address these two challenges.

The remainder of this paper is organised as follows: Section 2 provides an overview of the data source, which forms the foundation of this study. Section 3 presents the methodology, covering aspects of machine learning algorithms, structural complexity, and explainable techniques. Section 4 discusses the diagnosis and interpretation results, following the order outlined in the methodology. Section 5 offers a further discussion of the obtained results and outlines potential directions for future work. Finally, Section 6 concludes the paper.

2. Background

2.1. Data Source

The work presented in this paper is based on a simplified fuel rig designed to represent the Boeing 777 fuel system. Figure 1 illustrates the schematic of the fuel rig, with fuel flow moving from left to right, powered by a pump. Direct proportional valves (DPVs) were employed to accurately simulate degradation within the system, with DPV1 to DPV5 used to introduce and control five distinct failure modes: pump external leakage, pump internal leakage, FOHE (fuel oil heat exchanger) blockage, FOHE leakage, and clogged nozzle. Pressure sensors monitor the system's pressure, while flow meters track the flow rate.

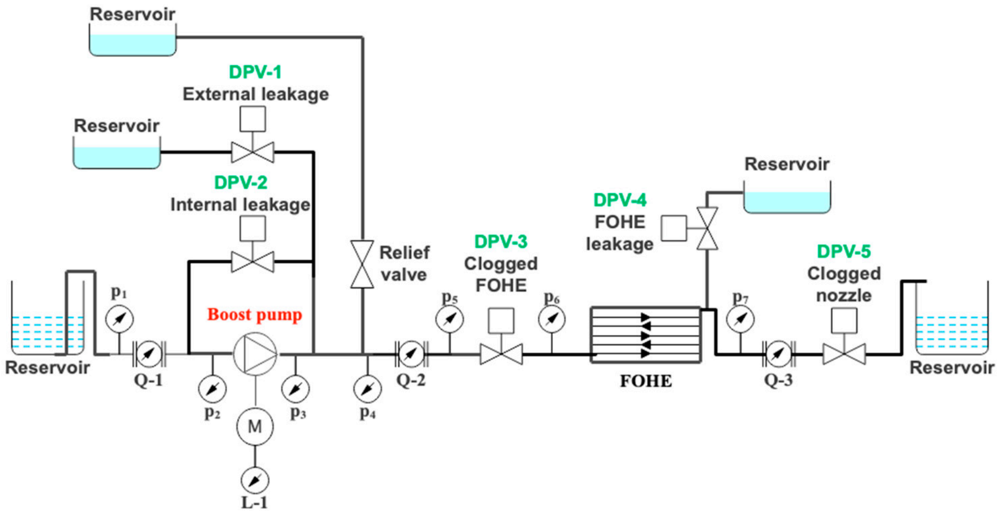


Figure 1. The schematic of the fuel rig.

As part of another task within this PhD project, a Simulink model (Figure 2) was developed using experimental data to create a digital twin of the fuel rig. Similar in function to the fuel rig, this model simulates five failure modes and their combinations under five specific pump speeds. Although this simulation work is beyond the scope of the current paper, the model has been systematically validated under both healthy and faulty conditions, with a maximum deviation of 3% between simulation results and experimental data. A more detailed discussion of this work will be presented in a subsequent article, which is currently under preparation. Consequently, the model is employed here to provide the necessary data for training, testing, and validating the ML-based diagnostic algorithms.

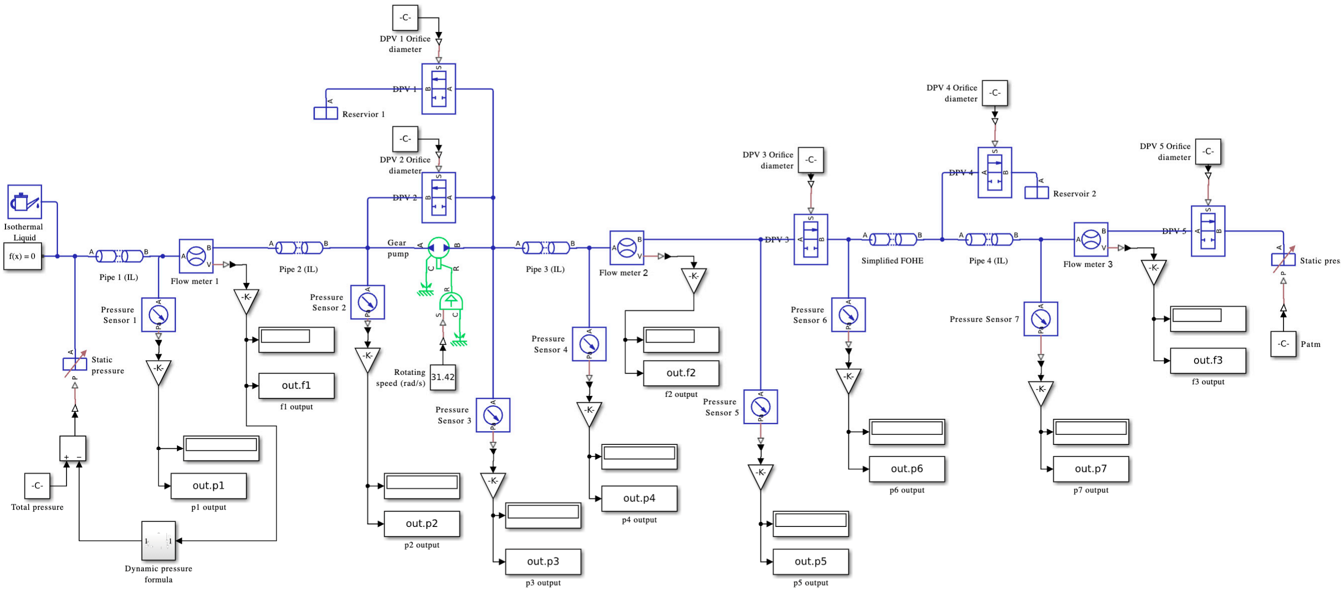


Figure 2. The Simscape model.

2.2. Data Preparation

This paper considers five failure modes within the aircraft fuel system (AFS), with their degradation severity outlined in Table 1. Each failure mode was categorised into five levels of severity, including a healthy condition and four progressively degraded states. The degradation process begins at 0% (indicating a completely healthy state) and progresses through specific intervals of degradation. The healthy and early degradation levels are considered tolerable (normal condition), while further degradation leads to functional failure (faulty condition), at which point the affected component requires repair or replacement. The boundaries between normal and faulty conditions were determined based on the impact of each failure mode on the performance metrics of static pressure and flow rate delivered to the engine (Figure 3). The red bars in the figure highlight the severity levels that significantly affect one or both performance metrics, thereby indicating faulty conditions (or functional failures).

Table 1. Degradation severities of each fault.

Failure Modes	Normal Conditions	Faulty Conditions
Pump ext leak	0%, 10%, 20%, 30%	40%
Pump int leak	0%, 10%, 20%	30%, 40%
FOHE block	0%, 10%, 20%	30%, 40%
FOHE leak	0%, 15%, 30%, 45%	60%
Nozzle block	0%, 10%, 20%	30%, 40%

In this simplified aircraft fuel system (AFS), the five failure modes can occur independently, allowing for simultaneous occurrences of multiple (0 to 5) faults. As mentioned in Section 1, this study aims to develop an end-to-end multi-fault diagnostic algorithm without relying on ensemble methods. Specifically, the approach involves feeding a single ML algorithm with a dataset to identify all potential faults in the system. To achieve this, the work considers all possible combinations of the failure modes. Consequently, this method innovatively transforms a typical multi-label problem, which often requires multiple algorithms to be addressed, into a multiclass problem solvable by a single algorithm. These five failure modes, each with two possible states (normal and faulty), formed 32 (2^5) multi-fault classes, comprising one normal class and 31 faulty classes. Table 2 provides the failure modes included in each class: Class 1 represents the normal class with all components functioning correctly, while Class 32 represents the worst-case scenario, where all components experience failure. Data were collected across all 32 classes, with the simulation model run for each possible combination of degradation severity. During each run, the pump speed was set to one of five fixed values (200, 300, 400, 500, and 600 rpm) to simulate different operating conditions. The final two rows of Table 2 show the minimum number of simulation runs needed to generate sufficient data across these conditions.

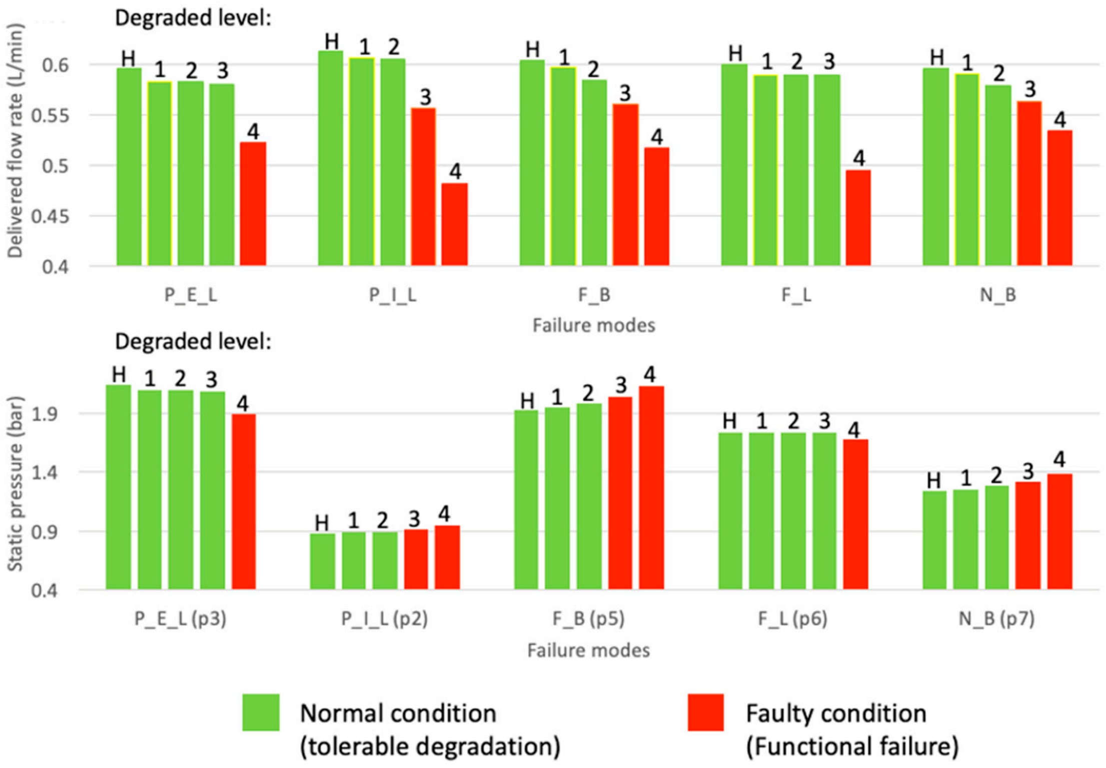


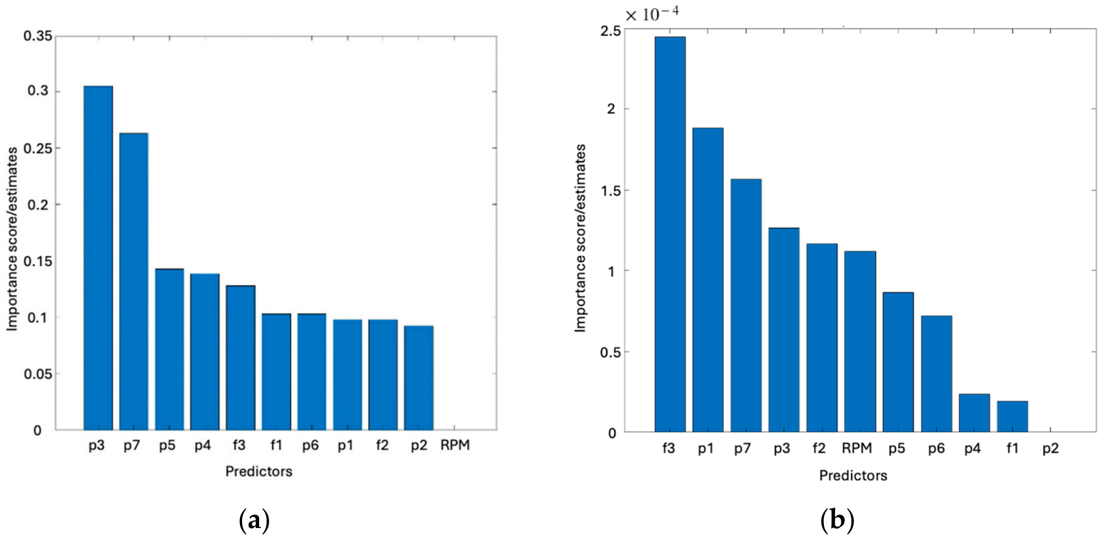
Figure 3. Failure modes' impact on the system's behaviour.

Each dataset comprised readings from eleven sensors: seven pressure sensors, three flow meters, and a laser sensor for measuring pump speed. Two feature selection algorithms, MRMR and tree-based methods, were applied to rank the importance of these features (Figure 4) in predicting the aforementioned 32 classes. However, the results of the two methods differed significantly. As a result, all 11 sensor outputs were selected to serve as 11 features with equal importance for training the ML algorithms. Pressure sensors were strategically placed throughout the system to capture pressure levels before and after key components, while the flow meters were positioned in series along the main pipeline: the first measured the total inflow from the fuel tank, the second monitored the pump's discharge flow downstream, and the third tracked the flow delivered to the engine. All data were normalised and rescaled to the range [0, 1] to ensure a balanced feature contribution in the ML models.

Table 2. All classes and their corresponding failure modes and number of runs.

	Normal class	**Faulty classes**																														
Failure modes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Pump ext leak																	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pump int leak									✓	✓	✓	✓	✓	✓	✓	✓									✓	✓	✓	✓	✓	✓	✓	✓
FOHE block					✓	✓	✓	✓					✓	✓	✓	✓					✓	✓	✓	✓					✓	✓	✓	✓
FOHE leak			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓
Nozzle block		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Number of runs (with 5 pump speeds)	2160	1440	540	360	1440	960	360	240	1440	960	360	240	960	640	240	160	540	360	135	90	360	240	90	60	360	240	90	60	240	160	60	40
Sum	15,625																															

✓ indicates fault modes included in each faulty class.

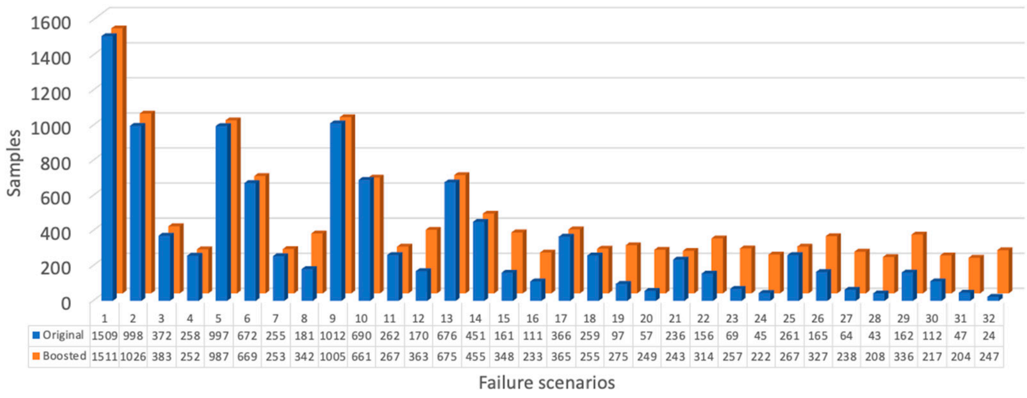


(**a**) (**b**)

Figure 4. Feature importance rankings: (**a**) minimum redundancy maximum relevance; (**b**) tree-based method.

The data for the algorithm development were randomly split into three subsets: training, testing, and validation. During training, 70% of the data were used to train the model, while 10% served as the testing data to evaluate and optimise hyperparameters, helping to find the best combination and reduce overfitting. The remaining 20% was set aside as validation data for a final, unbiased evaluation of the algorithm's performance.

To assess whether each class had sufficient data to train the diagnostic algorithm using these 11 features, the training dataset (comprising 10,938 instances) was categorised by class, with the distribution shown by the blue bars in Figure 5. As depicted, Class 1 contained a substantial amount of data. However, as the number of faults increased, the amount of available training data decreased correspondingly. For instance, when all components were in a failure state (Class 32), only 24 instances were available in the training data.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
■ Original	1509	998	372	258	997	672	255	181	1012	690	262	170	676	451	161	111	366	259	97	57	236	156	69	45	261	165	64	43	162	112	47	24
■ Boosted	1511	1026	383	252	987	669	253	342	1005	661	267	363	675	455	348	233	365	255	275	249	243	314	257	222	267	327	238	208	336	217	204	247

Figure 5. Distribution of the training data.

In contrast to experimental methods, the simulation model offered an efficient way to generate additional data for classes with fewer data points. To ensure adequate training, a minimum of 200 datasets was deemed necessary per faulty class. The simulation model was rerun for data-deficient classes, introducing random measurement uncertainties (noise) to create 3880 new samples. This noise, primarily originating from atmospheric pressure and sensor readings, followed a Gaussian distribution with distinct standard deviations: atmospheric pressure noise was set to a standard deviation of 500 Pa, based on Met Office data, while each sensor's noise standard deviation varied with pump speed, as derived from experimental data. This expanded "boosted dataset" contained a total of 19,505 datasets. Consistent with previous splits, 70% of these data (13,650 sets) were designated as new training data, with the class distribution shown in Figure 5 (red bar). This approach highlights the effectiveness of simulation techniques in rapidly generating large, diverse datasets compared to traditional experimental methods.

3. Methodology

3.1. Selected Machine Learning Algorithms

While deep learning is increasingly used for fault diagnosis and maintenance in the aviation industry, traditional machine learning (ML) remains essential, particularly in scenarios with limited data, straightforward features, high interpretability needs, or restricted computational resources. ML-based fault diagnosis relies on algorithms that use mathematical models to map data to fault classes directly, enabling data-driven decisions without needing prior domain knowledge or expert input. These simpler algorithms are often more interpretable, making them especially suitable for high-value assets like aircraft, where understanding model decisions is crucial for trust and safety. Consequently, ML's interpretability advantage over deep learning makes it valuable for fault diagnosis in aviation.

Given the limited sample size per class (around 200 samples) and the use of simple features with clear physical meanings, traditional ML was considered suitable for the fault diagnosis task in this study. Based on the data's characteristics, four criteria were established to guide the selection of appropriate ML algorithms:

- Classification task: this study frames fault diagnosis as a classification problem with discrete fault classes. Thus, all selected algorithms must be suitable for classification tasks.
- Supervised learning: the dataset from the simulation model is fully labelled, so only supervised learning methods were considered.
- Data efficiency and interpretability: unlike deep learning methods, the algorithms should perform well on small datasets and avoid excessive opacity to support ease of understanding and trust.
- Comparable predictive and interpretability performance: the chosen algorithms should balance predictive accuracy and interpretability, allowing for meaningful analysis and discussion of these key performance metrics.

These requirements ensure that the selected ML methods are effective for the available data and meet the demands for interpretability in the high-stakes context of aviation fault diagnosis.

Following the outlined requirements, logistic regressions, decision trees, support vector machines, and artificial neural networks were selected to evaluate diagnostic capability and interpretability in this study. Each method meets the data requirements and offers a balance between accuracy and interpretability. The following four subsections provide a concise overview of each approach.

The logistic regression (LR) is a foundational machine learning algorithm for classification, typically used in binary classification tasks where only two outcomes are possible. In a binary setting, such as distinguishing between circles and squares in Figure 6a, an LR estimates the likelihood of an event occurring based on the data. It uses a linear function to fit the data, passing through a nonlinear sigmoid function that maps the results to a range between 0 and 1. By setting a threshold, an LR classifies the data into two classes.

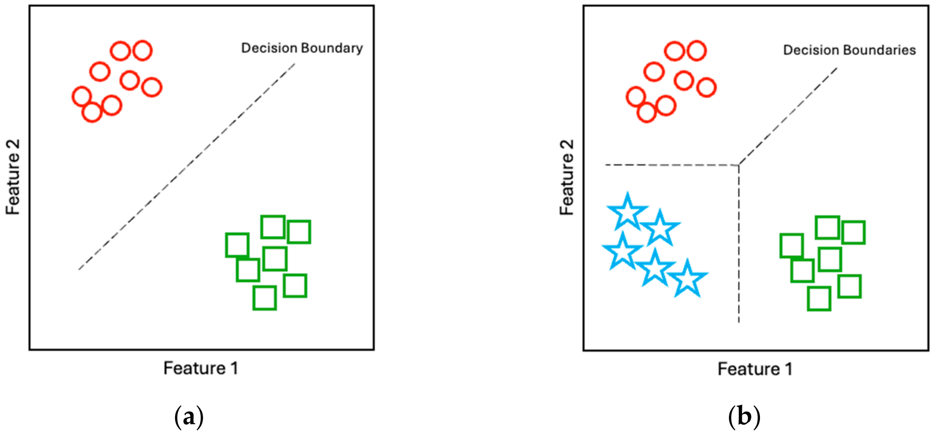


Figure 6. Examples of binary and multiclass classification: (**a**) binary class classification; (**b**) multiclass classification.

LRs can be extended to multinomial logistic regression for multiclass classification, where the data have more than two classes (e.g., circles, squares, and stars in Figure 6b). This approach applies multiple LR models to calculate the probability of each class relative to a reference class. Since the total probability across all classes must be 1, the algorithm predicts the class with the highest calculated probability. Multinomial logistic regression is, thus, well-suited for multiclass tasks. Although logistic regression has been infrequently applied to fault diagnosis for fuel systems, it was used in one study [25] to detect faults, underscoring its potential applicability.

Decision trees (DTs) are machine learning algorithms that represent decisions in a tree-like structure with multiple layers of nodes. Typically, a DT has at least two layers: the first to the $(n-1)$th layers comprises nodes where features are assessed, and branches represent decision outcomes. The nth layer consists of leaf nodes that indicate the classification results. A decision tree prioritises features for splitting based on information gain, favouring those that maximise this criterion.

Several common decision tree algorithms include ID3 ([26]), C4.5 ([27]), and CART ([28]). Unlike ID3 and C4.5, CART (classification and regression trees) limits each node to two branches and is applicable to both classification and regression tasks. CART often demonstrates better classification performance than other DT algorithms, though it can be limited by its reliance on local optimal choices, potentially impacting overall accuracy. In addition to binary classification, CART can handle multiclass classification, where leaf nodes correspond to distinct classes. Decision trees have been used effectively for fault detection and isolation in various contexts, including fuel systems [29], turbofan engines [30], and auxiliary power units [31].

Support Vector Machines (SVMs) are binary classification algorithms that classify data by constructing hyperplanes, maximising the margin between classes for improved separation. Given a dataset with m features, SVMs often apply a kernel function to map data into an $(m+n)$-dimensional space, allowing for linear separability in cases where the original data are not linearly separable. The dimension n depends on data complexity; if the data are inherently linearly separable, $n = 0$. The hyperplane for classification in this new space is defined by the closest data points, known as support vectors.

To handle multiclass classification (e.g., with k classes), SVMs typically employ two strategies: one-vs-all and one-vs-one. In the one-vs-all approach, k binary classifiers are trained, each distinguishing one class from all others. Each classifier outputs a score, and the class with the highest score is selected to avoid ambiguous classifications. The one-vs-one approach, on the other hand, creates an SVM model for each pair of classes, resulting in $k(k-1)/2$ models, with predictions determined by a majority vote among them. Although more computationally intensive, one-vs-one generally outperforms one-vs-all by better managing data imbalances in sub-classifiers. The studies in [6,32–34] highlight SVMs' effectiveness in fault detection for fuel systems.

Artificial neural networks (ANNs) with one hidden layer, or shallow networks, are composed of an input layer, a hidden layer, and an output layer, each containing neurons (or nodes) that connect fully with the adjacent layers. These connections carry weights that the ANN adjusts during training to fit the data. Since weights and biases (extra inputs to neurons) alone form linear operations, nonlinear activation functions are applied to the hidden and output layers to enable more complex, adaptable prediction capabilities. Common activation functions in the output layer include the sigmoid for binary classification or regression and the SoftMax function for multiclass classification.

While shallow ANNs are simpler than deep networks, they still exhibit black-box characteristics due to the hundreds of weighted connections they generate to learn patterns across features and classes, such as the 11 features and 32 fault classes in this study's AFS case. Weighted connections represent the network's assessment of the importance of relationships between neurons, which can aid in analysing simple neural networks. However, for a network aiming to fit 11 features and 32 classes, the large number of weighted connections reduces the efficiency and reliability of such analyses. Additionally, nonlinear activations further complicate the network's internal decision logic, making it challenging to interpret its classification processes. Nevertheless, shallow ANNs have shown promise in diagnosing mechanical faults, as demonstrated in prior studies [35–38].

3.2. The Structural Complexity of Selected ML Algorithms

A novel set of dimensionless metrics based on internal connections or weights was introduced to enable a fair comparison among machine learning algorithms with varying structural complexities. This approach balances the evaluation, accounting for each algorithm's unique structural intricacies. Here is how structural complexity was defined for each algorithm:

LR: for multinomial logistic regressions, each class prediction relies on weights associated with input features and an additional bias (or residual) term for each class. Thus, the total structural complexity is determined by the number of weights and residuals required to compute probabilities across all classes.

DT: decision trees partition the data using split nodes at each layer. Each split node evaluates one feature to split the data. Therefore, the model's structural complexity correlates with the number of split nodes. More nodes allow for finer granularity in the classification, enhancing performance but increasing complexity.

SVM: the complexity of an SVM in multiclass classification is determined by the number of binary sub-classifiers required. Based on the discussion above, two ensemble learning strategies, one-vs-one and one-vs-all, are commonly employed in multiclass problems. These strategies result in different levels of complexity, corresponding to $\mathrm{k}(\mathrm{k}-1)/2$ and k, respectively, where k is the total number of classes.

ANN: for ANNs, structural complexity is straightforwardly determined by the number of connections, or weights, within the network. For a single-hidden-layer ANN with x input nodes, y hidden nodes, and z output nodes, the total number of weights is calculated as: $(x \times y) + (y \times z)$. This includes the connections between the input layer and the hidden layer and those between the hidden layer and the output layer. This measure reflects the model's ability to capture intricate patterns in the data, with higher complexity associated with larger networks.

This complexity-based framework allows for a balanced assessment of model performance by normalising their predictive power in relation to their structural demands. This makes comparisons meaningful despite inherent differences in algorithm design.

3.3. Evaluating the Interpretability of the Selected ML Algorithms

Overview of the proposed validation process. Figure 7 illustrates an innovative structured process proposed to evaluate, quantify, and compare the interpretability of four ML algorithms by validating their explanation outputs against domain knowledge. Interpretability metrics were derived through two distinct methods designed for transparent and black-box algorithms. Domain knowledge, encompassing residuals and key features, serves as a benchmark for assessing the alignment between algorithmic outputs and engineering insights.

Method 1: interpretability assessment for transparent algorithms. Method 1 was applied to transparent ML algorithms, such as linear regressions (LRs) and decision trees (DTs), where model components inherently lend themselves to interpretation. Here, a residual table (Table 3) was employed to validate model interpretations. Interpretability stems from analysing feature coefficients for LRs, while it originates from diagnostic rules based on feature splits for DTs. The alignment between residual data and model outputs establishes the interpretability of these transparent algorithms.

Method 2: evaluating interpretability of black-box algorithms using XAI techniques. For black-box algorithms, interpretability was assessed using explainable artificial intelligence (XAI) techniques—specifically LIME and SHAP, both local and model-agnostic methods. These techniques facilitate an analysis of key features identified by XAI as significant to model predictions. A key-feature table (Table 4) provides an engineering basis to evaluate XAI outputs, with interpretability measured by the degree of overlap between the XAI-derived sensitive features and those identified as critical in engineering analysis. This approach has proven to be effective in applications such as unmanned aerial vehicle (UAV) elevators [19], gas turbines [39], nuclear power plants [40], cross-building energy systems [41], and rotating machines [42], where XAI techniques interpret the diagnostic logic of complex ML models.

Symptom vector formation. The process, shown in the left half of Figure 7, began with constructing symptom vectors based on pressure (p1 to p7) and flow rate (f1 to f3) features within the dataset. These features represent key indicators of system health, with each condition—such as normal operation, pump external leakage, or fuel oil heat exchanger (FOHE) blockage—having its own symptom vector (shown in Figure 8). A matrix was created for each of the five pump speeds, with rows representing classes and columns representing features. The mean values for each class and pump speed were then recorded in the matrix, providing a baseline for subsequent analysis.

Residual calculation and interpretation. Residuals from the symptom vectors produced a 31 × 10 matrix for each pump speed, highlighting deviations between each faulty class and the normal condition. For example, Table 3 records the residuals for nozzle blockages (Class 2). Engineering knowledge interprets the resulting backpressure as an increase in downstream pressure (p3 to p7) and a reduction in flow rate (f1 to f3). Such correlations between engineering principles and residual data underpin the interpretability of diagnostic models.

Key feature identification for fault diagnosis. Analysing the residuals and leveraging engineering knowledge can identify the key features most indicative of each faulty class. These features show the greatest deviation from the normal condition and are instrumental in diagnosing specific faulty classes. Table 4 shows the key features for several faulty classes, highlighting the four features most significantly deviating from the normal condition. These key features are crucial for diagnostic accuracy and serve as the foundation for validating the interpretability of the black-box algorithms.

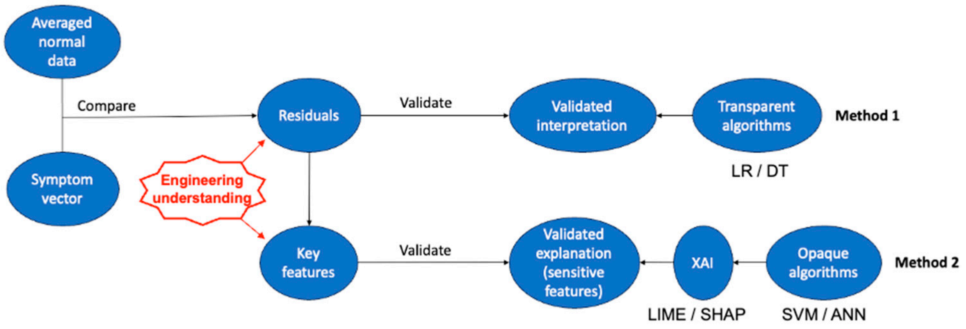


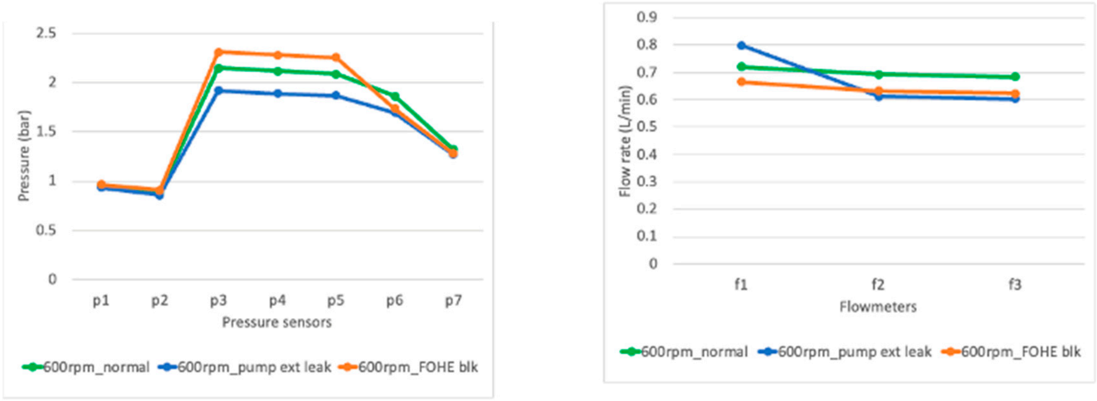
Figure 7. The process of validating explanations with engineering understanding.

Table 3. The difference between faulty classes and the normal class.

Class	p1	p2	p3	p4	p5	p6	p7	f1	f2	f3
2	+	+	+	+	+	+	+	−	−	−
3	−	−	−	−	−	−	−	+	+	−
4	−	−	−	−	−	−	−	+	+	−
5	+	+	+	+	+	−	−	−	−	−
6	+	+	+	+	+	−	+	−	−	−
7	+	+	+	+	+	−	−	−	−	−

Table 4. Key features of faulty classes.

Class	Key Features			
2	p7	p6	p5	p4
3	f3	p7	p6	p1
4	f3	p7	p6	f2
5	p5	p4	f3	f2
6	p5	p7	f3	p4



(**a**) (**b**)

Figure 8. Symptom vectors of three health conditions: (**a**) pressure profiles; (**b**) flow rate profiles.

4. Results

4.1. Multi-Fault Diagnostic Results

In summary, the simplified AFS model comprised five failure modes, producing 32 distinct classes—one representing normal conditions and 31 representing various faulty conditions, including multiple faults. This dataset included 19,505 individual datasets. The objective of multiple-fault diagnosis is to develop an algorithm that accurately identifies all 32 classes. The performance of the selected ML algorithms was evaluated using the following metrics: multiclass confusion matrix, accuracy, Matthews correlation coefficient (MCC), kappa coefficient, and F-1 score.

Before finalising each algorithm's configuration, including model structure and hyperparameters, preliminary evaluations were performed using an independent test set comprising 10% of the total dataset, instead of cross-validation. This approach was adopted due to the ample data generated by the simulation model for ML. The entire dataset was randomly divided into three subsets—training, testing, and validation—ensuring that each subset had a similar distribution and included all 32 classes. Figure 9 highlights the test set using the yellow color and illustrates the quantity of data for each class within it.

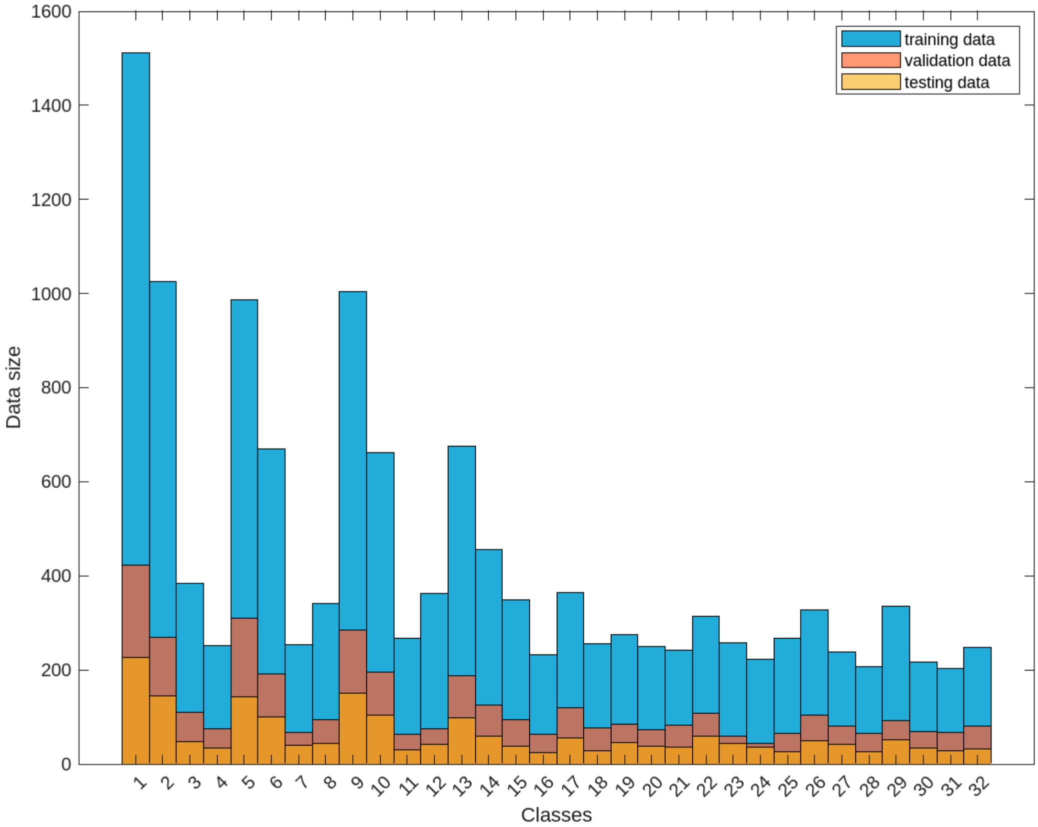
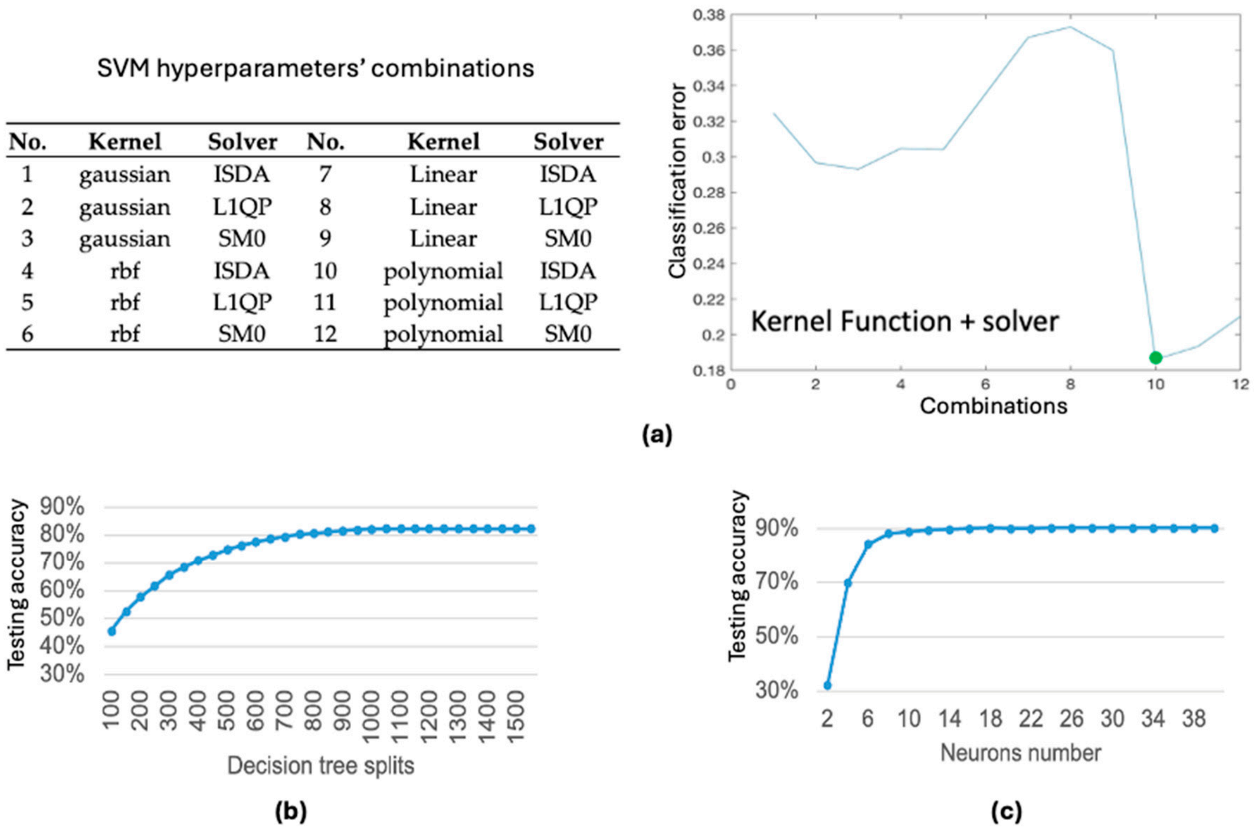


Figure 9. Distribution of training, testing, and validation datasets.

A straightforward approach was then employed in the MATLAB R2024a environment to tune only one to two hyperparameters that significantly impacted algorithm performance based on accuracy with the test data. All other parameters were kept at their default values. For the logistic regression, no hyperparameters were identified as substantially affecting performance. In the case of the SVM, the choice of kernel functions and solvers significantly influenced performance, with MATLAB offering four kernel options and three solver options, as summarised in Figure 10a. The key hyperparameters for decision trees (DTs) and artificial neural networks (ANNs) were the number of decision nodes and neurons in the hidden layers, respectively.



SVM hyperparameters' combinations

No.	Kernel	Solver	No.	Kernel	Solver
1	gaussian	ISDA	7	Linear	ISDA
2	gaussian	L1QP	8	Linear	L1QP
3	gaussian	SM0	9	Linear	SM0
4	rbf	ISDA	10	polynomial	ISDA
5	rbf	L1QP	11	polynomial	L1QP
6	rbf	SM0	12	polynomial	SM0

Figure 10. ML results on the test data: (**a**) SVM testing results; (**b**) DT testing results; (**c**) ANN testing results.

Finally, based on the classification errors under the different hyperparameter settings shown in Figure 10, a one-vs-one strategy was implemented for the SVM. This configuration used a 4th-order polynomial kernel, 496 sub-learners, and an ISDA solver, which yielded the lowest classification error. Additionally, when achieving the highest test accuracy, the ANN featured a three-layer network with 11 input nodes, a hidden layer of 16 nodes, and 32 output nodes. The decision tree (DT) utilised a CART structure with 1100 splits (decision nodes).

Subsequently, the optimised models were evaluated on the remaining validation dataset, representing 20% of the total data. Their performance was assessed using the four evaluation metrics previously mentioned. The results of this validation process are summarised in Table 5.

Table 5. Performance on the validation dataset.

	Accuracy	MCC	Kappa	F-1
LR	0.7926	0.7826	0.781	0.7535
DT	0.8382	0.8303	0.83	0.7874
SVM	0.8395	0.8319	0.8308	0.8195
ANN	0.9008	0.8959	0.8958	0.879

All four evaluation metrics indicate that the ANN outperformed the other models, correctly classifying 90% of the samples, while the logistic regression (LR) showed the weakest performance. Figure 11 provides a detailed view of the F-1 score distribution across classes for the four selected ML methods, with the F-1 score being the only metric capable of offering class-level detail. The average F-1 scores in Table 5 correspond to the values in Figure 11. The ANN achieved the highest F-1 scores in nearly all classes except for Class 12 and 17. Conversely, the LR exhibited the lowest F-1 scores in most classes, with only seven exceptions. The F-1 scores of the SVM and DT generally fell between those of the ANN and LR, although the DT's average score was notably lower than that of the SVM due to higher variability across classes. Specifically, the DT's performance was particularly poor in certain classes, including Classes 11, 15, 16, 30, and 31.

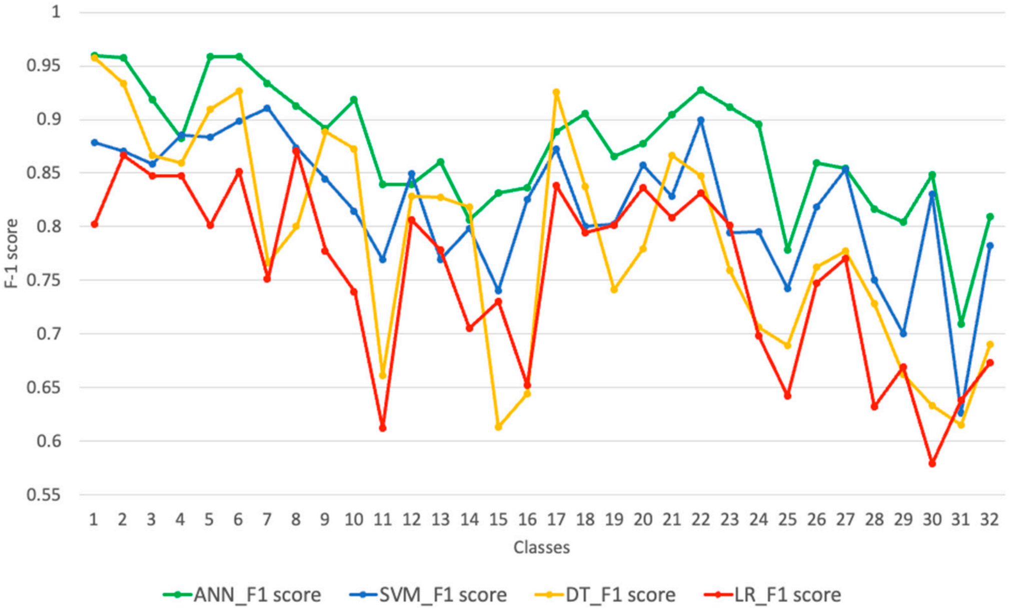


Figure 11. F-1 score of the four algorithms across the 32 classes.

Figure 12 presents the confusion matrix for SVM results on the validation dataset. In this matrix, rows correspond to the actual classes, while columns represent the SVM's predicted classes. Correct predictions are highlighted in blue, and incorrect predictions are highlighted in red. The color intensity indicates the number of cases in each cell.

The distribution in Figure 12 reveals that SVM predictions were concentrated along several distinct lines, including incorrect predictions aligned parallel to the diagonal. Starting at row 9, column 1, the first red line represents misclassifications related to pump internal leakage, indicating that the SVM struggled to detect this failure mode. Similarly, the second red line reflects instances where the SVM failed to identify a pump external leakage. The third red line shows cases where the SVM did not consistently recognise simultaneous internal and external leakages in the pump. Notably, the confusion matrix for the ANN exhibited a similar pattern, with false negatives concentrated in these three fault classes.

Additionally, the logistic regression (LR) frequently produced false negatives for FOHE leakages, often misclassifying faulty cases as normal. In contrast, the decision tree (DT) model generated a high number of false positives for the same failure modes, potentially leading to increased maintenance costs. Among the failure modes in the simplified AFS, blockages were generally easier to detect than leakages. Overall, based on these analyses, the ANN and the SVM demonstrated a superior performance compared to the LR and the DT.

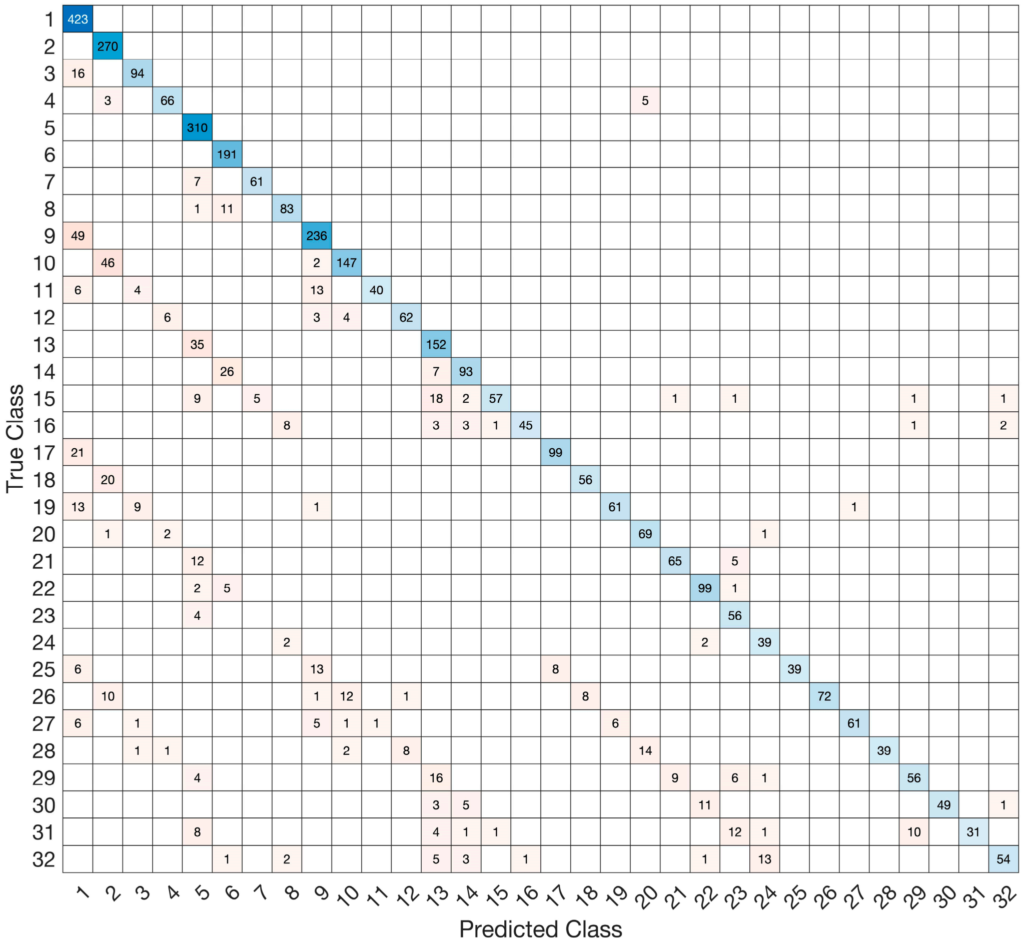


Figure 12. Confusion matrix of the SVM.

4.2. The Structural Complexity and Balanced Results

Following the method outlined in Section 3.2, the structural complexity of the four trained ML algorithms was quantified to facilitate fair comparisons. This process balances complexity across algorithms. For the multinomial logistic regression, the model predicts the class with the highest probability among 32 options. Each probability calculation requires 11 weights (one for each feature) and a bias term, resulting in 384 parameters, defining logistic regression's structural complexity. Adjustments were made to the other algorithms to ensure comparable complexity, such as reducing the number of decision nodes in the decision tree (DT) from 1100 to 400 and shrinking the number of hidden neurons in the artificial neural network (ANN) to nine (from 16).

It is important to note the structural complexity of the SVM, which employs a one-vs-one strategy and has a complexity of 496, corresponding to the number of sub-learners. However, to align with the simpler structure of the logistic regression, which only involves first-order calculations, the nonlinear kernel function order for the SVM was reduced from fourth to second order. Table 6 summarises the structural complexities of all four algorithms before and after the balancing adjustments.

Table 6. The structural complexity of each algorithm.

ML Algorithm	Original Complexity	Balanced Complexity
Logistic regression	384	384
Decision tree	1100	400
SVM	496	496
ANN	688	387

The four ML models were retrained using the adjusted configurations described above, and the validation results are presented in Table 7.

Table 7. Performance on the validation dataset after balancing complexity.

Algorithm	Accuracy	MCC	Kappa	F-1
LR	0.7877	0.7773	0.7757	0.7481
DT	0.7101	0.6955	0.6942	0.6351
SVM	0.7729	0.7622	0.7596	0.7398
ANN	0.8787	0.8727	0.8725	0.8538

The results indicate that the ANN demonstrated the highest predictive performance, while the DT showed the lowest. The SVM and logistic regression ranked second and third, respectively. Reducing model complexity negatively impacted performance, a fact which is unsurprising. However, achieving higher accuracy typically requires increasing model complexity, making the model more challenging to interpret.

4.3. Interpretability of the Trained Diagnostic Algorithms

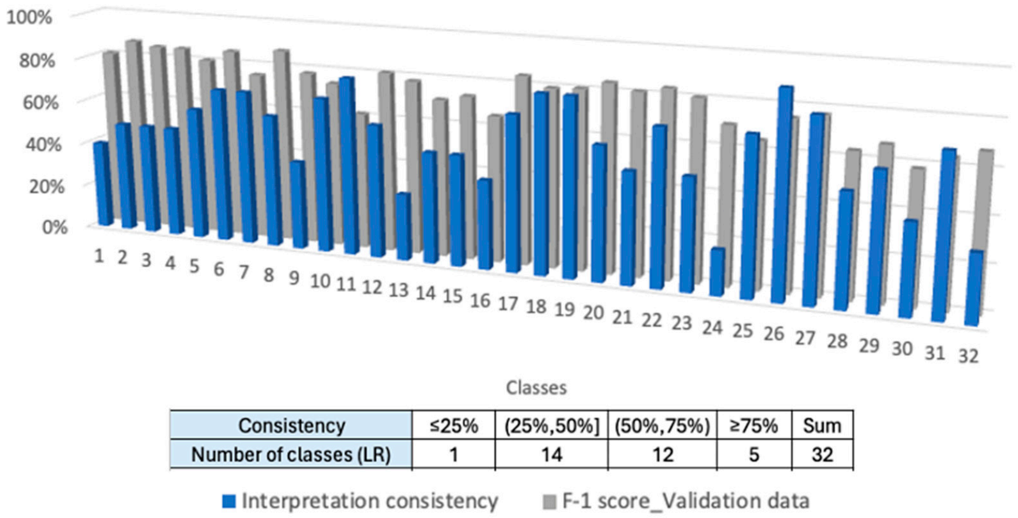
LR. In multinomial logistic regressions, model interpretation is based on the analysis of coefficients, representing each feature's contribution when comparing faulty classes to the normal class. A larger absolute coefficient value indicates a stronger influence of that feature on the prediction. Positive coefficients suggest a positive relationship between the feature and the likelihood of a specific fault class, while negative coefficients imply an inverse relationship.

Table 8 displays selected coefficients from the multinomial logistic regression model for Classes 2 to 16. Each column corresponds to a fault class, while each row represents the coefficients for a specific feature. For instance, in the case of FOHE blockages (Class 5 to 8), the negative coefficient for p6 indicates a negative relationship between p6 and the fault's occurrence. This suggests that a lower downstream pressure at p6 increases the likelihood of FOHE blockages, aligning with engineering principles. Similarly, the coefficients for pump speed and flow rate f1 indicate strong correlations with internal leakages (Class 9 to 16). According to the model, internal leakages become more probable as pump speed rises and the system flow rate decreases.

Table 8. Coefficients of the multinomial logistic regression.

Classes	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Intercept	6.54	−1.3	−3.27	8.03	8.56	−4.65	0.65	6.77	9.97	0.59	8.98	8.62	7.78	2.75	1.28
Pump speed	−3.35	2.24	−1.5	1.60	−1.1	0.69	−3.5	33.75	34.41	36.94	35.00	34.91	35.03	35.27	36.04
p1	−1.76	0.34	−1.24	−3.17	−5.45	4.42	−0.12	−0.64	−3.55	1.77	−2.67	−1.15	−3.98	4.01	−2.04
p2	−8.35	−0.84	2.06	−5.29	−4.75	−0.15	−1.52	−1.3	−3.2	3.36	−3.37	−1.47	1.32	−0.01	6.12
p3	1.45	−6.38	−5.21	8.25	2.90	4.11	7.64	0.54	−12.75	−5.82	−3.15	−0.97	6.61	0.14	3.68
p4	−8.3	1.72	3.83	22.33	25.16	12.57	22.33	−10.32	−1.15	6.07	−26.29	16.00	13.21	17.47	25.32
p5	1.56	4.06	−2.12	31.92	37.42	52.11	45.26	−33.25	−24.22	−24.79	−8.94	25.61	15.45	20.04	7.76
p6	−61.31	−31.75	−28.98	−84.49	−138.2	−84.12	−105.95	−1.21	−53.49	−9.31	−51.95	−82.55	−116.25	−86.48	−101.39
p7	87.27	4.41	53.57	6.88	101.44	9.33	71.71	8.34	83.51	−15.41	68.97	1.21	82.65	−22.64	78.98
f1	−18.41	−17.86	−10.17	−10.42	−25.24	−6.44	−24.52	−32.68	−46.68	−41.18	−42.01	−37.95	−40.83	−32.46	−44.76
f2	36.93	99.17	92.65	14.56	32.50	68.06	89.34	0.20	33.79	86.14	103.69	0.20	11.22	86.22	83.84
f3	−22.92	−68.55	−92.55	0.00	−20.95	−59.29	−89.02	9.86	−22.79	−69.28	−94.34	13.82	−11.75	−65.9	−93.56

In Figure 13, the horizontal axis represents the 32 classes, while the vertical axis shows the results as percentages. The blue bars represent the consistency between the signs of the logistic regression (LR) parameters fitted for each class (as shown in Table 8, representing the LR's interpretation) and the corresponding engineering insights in Table 3. The grey bars depict the F-1 scores of the LR on the validation dataset. The table below the graph summarises the distribution of the consistency levels across classes. This consistency varied significantly among classes and was low (≤50%) in nearly half of them, resulting in an average interpretability of only 58% for the LR. Additionally, the LR's predictive performance, shown by the grey bars, was weaker than that of the other three algorithms, as indicated in Figure 11.



Consistency	≤25%	(25%,50%]	(50%,75%)	≥75%	Sum
Number of classes (LR)	1	14	12	5	32

Figure 13. The interpretability of multinomial logistic regression.

DT. Decision tree (DT) results are generally intuitive and straightforward to interpret. The decision rules for each class can be traced by following the branches and nodes from the root to the leaf nodes. However, as the tree structure expands, interpretation becomes increasingly complex. Despite this, specific rules can be extracted by focusing on individual paths from the root node to each leaf node.

In this study, the DT algorithm required 1100 decision nodes across more than 20 layers, with over 900 leaf nodes to classify all 32 classes effectively. Table 9 provides an example of a rule extracted from the trained DT. This rule involves 23 decision points, meaning that the corresponding branch contains 24 nodes: one root node, 22 internal nodes, and one leaf node. The table outlines each node in this branch except for the leaf node. The second column specifies the features used by the DT at each decision point, while the fourth column details the decisions made at these nodes. For continuous features (e.g., pressure and flow rate), the decisions are based on split points (fifth column). The decisions indicate specific feature values for discrete features (e.g., pump speed).

Table 9. Example rule extracted from the trained DT.

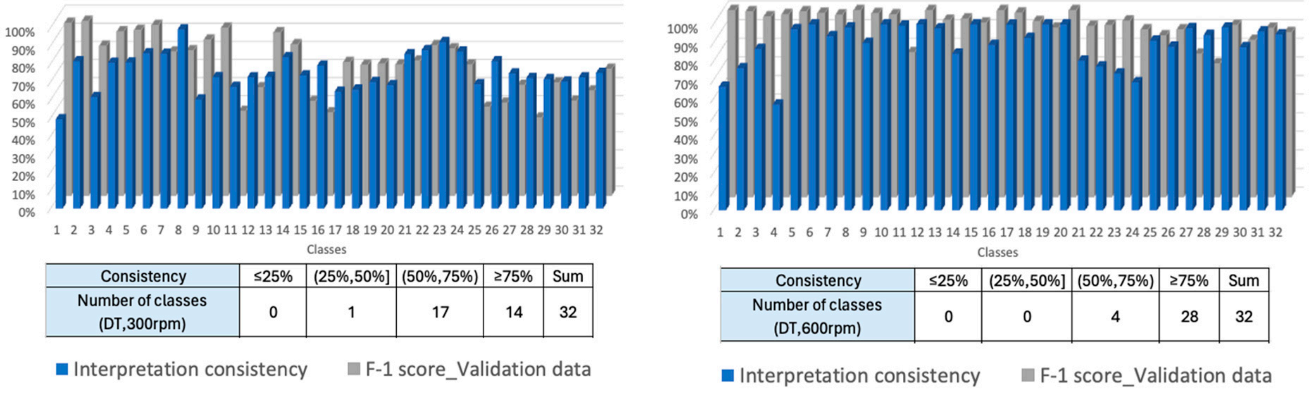
No.	Parent Node	Predictor	Type	Judge	Criteria	Unit
1	1	p7	Continuous	Less than	1.3837	bar
2	2	f3	Continuous	Less than	0.6596	L/min
3	4	p4	Continuous	Less than	2.1767	bar
4	8	f3	Continuous	Less than	0.5706	L/min
5	16	p7	Continuous	Less than	1.2293	bar
6	22	p5	Continuous	Less than	1.6392	bar
7	34	f3	Continuous	Less than	0.466	L/min
8	56	f3	Continuous	Equal or larger than	0.3643	L/min
9	77	Pump	Categorical	300	NaN	RPM

Table 9. *Cont.*

No.	Parent Node	Predictor	Type	Judge	Criteria	Unit
10	102	p6	Continuous	Less than	1.3799	bar
11	142	p3	Continuous	Equal or larger than	1.4321	bar
12	209	p6	Continuous	Equal or larger than	1.334	bar
13	305	p1	Continuous	Equal or larger than	0.9944	bar
14	429	p3	Continuous	Less than	1.4559	bar
15	596	f1	Continuous	Equal or larger than	0.3938	L/min
16	809	f3	Continuous	Equal or larger than	0.3707	L/min
17	1027	p5	Continuous	Less than	1.4101	bar
18	1204	f1	Continuous	Less than	0.4035	L/min
19	1392	p3	Continuous	Equal or larger than	1.4343	bar
20	1577	p1	Continuous	Less than	0.9952	bar
21	1732	p3	Continuous	Less than	1.4433	bar
22	1826	p3	Continuous	Less than	1.4397	bar
23	1876	f1	Continuous	Less than	0.4001	L/min

The trained DT contained 939 rules (leaf nodes), with rule counts varying across different target classes. Table 10 shows the number of rules associated with specific classes at given pump speeds. Notably, the DT generally required more rules for classifications at lower pump speeds (e.g., 200 rpm) than at higher pump speeds.

A comparison between the features associated with DT rules and the engineering insights (Table 3) is illustrated in Figure 14. This figure shows the alignment between the DT's rules and engineering perspectives (blue bars) alongside the F-1 score on the validation dataset (grey bars) at 300 and 600 rpm pump speeds. The interpretability of the DT is summarised in the tables below each graph. At the lowest pump speed of 300 rpm (Figure 14a), there were some discrepancies between the DT's interpretations and engineering understanding. However, the interpretability at 300 rpm was notably higher than that of the logistic regression, as indicated by the table statistics. At the highest pump speed of 600 rpm (Figure 14b), both the interpretability and predictive performance of the DT improved, resulting in an average interpretability of 82% across all five pump speeds.



Consistency	≤25%	(25%,50%]	(50%,75%)	≥75%	Sum
Number of classes (DT,300rpm)	0	1	17	14	32

Consistency	≤25%	(25%,50%]	(50%,75%)	≥75%	Sum
Number of classes (DT,600rpm)	0	0	4	28	32

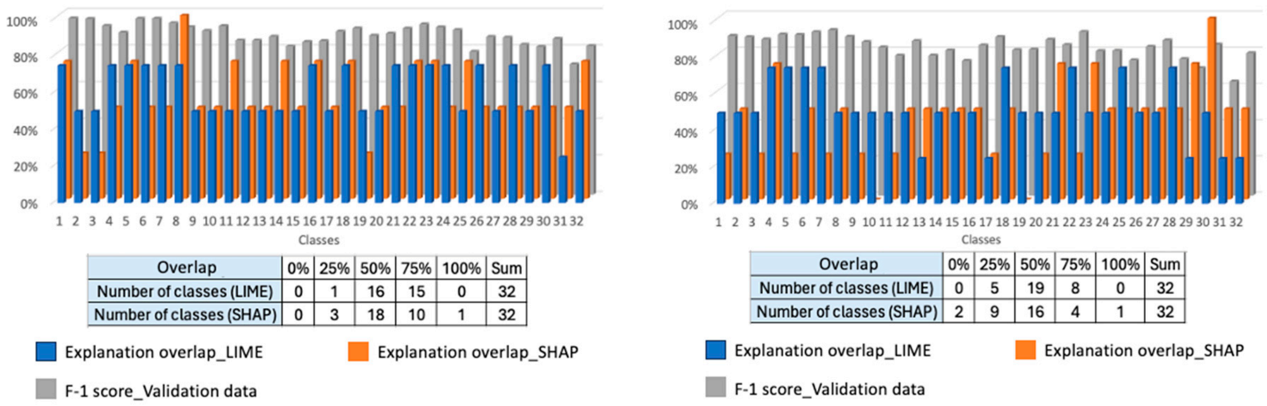
(**a**) (**b**)

Figure 14. The interpretability of the decision tree under different pump speeds: (**a**) RPM = 300; (**b**) RPM = 600.

Table 10. Distribution of rules across classes and pump speeds.

Class	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
200 rpm	18	9	15	6	4	7	3	10	12	3	6	17	9	9	8	7	13	8	11	7	4	8	8	5	3	16	8	4	9	9	8	6
300 rpm	10	7	6	6	10	3	8	5	15	12	12	9	13	10	13	10	4	5	7	5	5	5	3	6	10	9	9	7	7	10	8	8
400 rpm	6	4	1	4	6	5	7	5	7	8	7	6	6	9	8	9	2	2	3	4	4	5	6	5	6	6	9	8	9	7	6	3
500 rpm	3	6	3	3	8	3	5	3	6	1	2	1	8	10	5	6	2	2	2	1	2	2	3	6	4	4	5	4	4	5	6	4
600 rpm	3	2	2	3	3	3	4	2	2	2	4	1	4	4	6	3	1	4	3	1	5	2	2	3	5	4	5	3	4	4	2	4
Sum	40	28	27	22	31	21	27	25	42	26	31	34	40	42	40	35	22	21	26	18	20	22	22	25	28	39	36	26	33	35	30	25
Total	939																															

SVM and ANN (black-box algorithms). The LIME and SHAP techniques were applied to identify the four features most influential for each prediction and to calculate the degree of overlap (expressed as a percentage) between the XAI outputs and the critical features recommended by engineering insights (Table 4). For instance, a 100% overlap indicates that all features identified by XAI match those suggested by engineering insights, while a 50% overlap signifies that only two features align. The results are visualised in Figure 15, categorised by different classes.



Overlap	0%	25%	50%	75%	100%	Sum
Number of classes (LIME)	0	1	16	15	0	32
Number of classes (SHAP)	0	3	18	10	1	32

Overlap	0%	25%	50%	75%	100%	Sum
Number of classes (LIME)	0	5	19	8	0	32
Number of classes (SHAP)	2	9	16	4	1	32

(**a**) (**b**)

Figure 15. The interpretability of two black-box algorithms: (**a**) ANN; (**b**) SVM.

In Figure 15, the blue and orange bars indicate the overlap between the outputs of LIME and SHAP and the engineering insights, representing the interpretability of these opaque algorithms. The grey bars depict the F-1 score of the ML algorithms on the validation dataset, effectively decomposing the overall F-1 score from Table 5 across individual classes. For instance, for Class 2 in the ANN which related to a clogged nozzle prediction, the F-1 score—a measure of predictive accuracy—approached 95%. However, only two and one of the four most sensitive features identified by LIME and SHAP aligned with engineering insights. Although the ANN generally performed well across classes, the interpretability provided by the XAI techniques varied significantly.

The table in Figure 15 summarises the number of classes at each level of overlap. Compared to the SVM (Figure 15b), the ANN (Figure 15a) had more classes with high interpretability, defined as an overlap of 75% or greater. Based on these tables, LIME provided slightly better interpretability than SHAP for the ANN and the SVM, achieving 61% and 52% overlap, respectively. Thus, among the combinations of XAI techniques and opaque algorithms in this study, LIME provided the highest interpretability with the ANN, while SHAP offered the lowest interpretability with the SVM.

LIME's results for the ANN and the SVM in Figure 15 exhibit an approximate bimodal pattern, indicating a notable gap between the XAI output and engineering insights for Classes 9 to 16 and 25 to 32. These classes share a common failure mode: pump internal leakage. Unlike the external leakage, internal leakage results in the leaked flow recirculating to the pump inlet rather than exiting the system. This backflow reduces the system's flow rate and the pressure differential across the pump, giving the impression of normal operation at lower pump speeds. Although the selected opaque ML algorithms can detect internal leakage by incorporating pump speed into the feature space (as shown by the grey bars in Figure 15), this failure mode challenges the surrogate model in LIME, reducing interpretability.

Based on the engineering insights into pump internal leakages, the pump speed is the most sensitive feature for detecting this failure mode, followed closely by the flow rate delivered to the engine (f3) which significantly decreases during an internal leakage. However, in the XAI explanations for the ANN, pump speed, f3, and p7 were most frequently highlighted as essential features for predicting internal leakages. While engineering insights support the relevance of pump speed and f3, they disagree on the importance of p7, as it is located farther from the pump and experiences less significant pressure changes than p2 or p3. For the SVM, XAI identified flow rate f3 and pressures p5 and p7 as the most sensitive features, but engineering understanding only supports f3. Consequently, XAI explanations for the SVM underperformed relative to the ANN in certain classes (e.g., Classes 9, 10, 11, 13, 29, 31, and 32) associated with internal leakage.

5. Discussion

This work addresses a more complex multi-fault diagnosis scenario compared to previous studies. It thoroughly investigates the capabilities of traditional machine learning methods, including the shallow neural network, in fault diagnosis. Additionally, this study uniquely examines the influence of algorithmic structural complexity on fault diagnosis outcomes. A more balanced and fair comparison was achieved by re-evaluating the performance of four algorithms under similar structural complexities. In their optimal configurations, the ANN demonstrated the best predictive performance, with an average accuracy of 90% on the validation dataset and strong F-1 scores across all 32 classes. The SVM achieved an accuracy of 84%, with the F-1 scores varying by class and reaching 100% accuracy in certain cases. Among the transparent algorithms, the DT had an overall accuracy of 83.8%, lower than that of both the SVM and the ANN, while the LR, as the simplest model, yielded the lowest accuracy at 79%. Predictive performance decreased as model complexity was reduced, yet the ANN maintained the highest accuracy (87.9%). Additionally, the opaque algorithms consistently outperformed the transparent algorithms in diagnostic accuracy.

For the failure modes considered here, leaks in the AFS presented a greater challenge for ML algorithms than blockages. This difficulty is reflected in the confusion matrix, where a notable number of missed detections and false positives are associated with leaks. A possible reason for this discrepancy is that blockages produce more pronounced changes in pressure and flow rate than leaks, making them easier for ML algorithms to identify from the data.

While black-box algorithms such as SVMs and ANNs exhibit strong diagnostic capabilities, they present interpretability challenges for domain users. To address this, this study applies XAI techniques to explain black-box models and leverages engineering insights into the fuel system to validate these explanations. The consistency between model outputs and engineering understanding serves as a measure of interpretability, where a higher degree of overlap indicates better interpretability. This approach is necessary because LIME and SHAP results are influenced by the scope of the input data, whereas engineering understanding—rooted in human experience and expertise—extends beyond the dataset.

Despite requiring more computational resources and time, SHAP provided interpretability levels similar to those of LIME. Specific to each ML model, the ANN achieved better interpretability than the SVM (61% vs. 52%). However, as shown in Figure 15, consistency between engineering judgment and feature relevance from both LIME and SHAP varied widely, with differences of up to 50% across classes. For transparent models, the logistic regression (LR) parameters in Table 8 differ substantially from engineering insights across many classes. The LR's parameters represent global decision-making across the training dataset which contrasts with the local explanations provided by LIME and SHAP.

While LRs offer intuitive global interpretability, their simplicity limits their interpretability (only 58%), restricting their ability to capture complex mappings fully.

In contrast, the DT achieved higher interpretability (82%) and displayed greater consistency across fault classes, especially at high pump speeds. DTs' interpretability advantage stems from their rule-based approach, which divides the feature space into distinct regions associated with different classes. Class data are dispersed at high pump speeds, reducing inter-class ambiguity and simplifying classification and diagnostic tasks.

These findings suggest a trade-off in performance outcomes depending on the emphasis on interpretability versus accuracy. The DT offered relatively consistent interpretability across fault classes but achieved moderate accuracy, whereas the ANN delivered superior diagnostic accuracy at the expense of interpretability.

Future research could focus on employing XAI methods with a higher fidelity to improve model explanations, incorporating deeper engineering insights from experienced practitioners and addressing dimensionality challenges by reducing the number of classes or features. Particularly for the latter direction, a potential approach involves investigating the failure mode ratio (FMR) of components. In failure modes and effects analysis (FMEA), FMR serves as a metric describing the frequency of different failure modes in a system or device during operation, helping to identify the primary (frequent) failure modes within the system. By leveraging this approach, future studies could prioritise high-frequency failure combinations, thereby streamlining the classification process and reducing the number of classes.

6. Conclusions

As a critical onboard system, the health of the aircraft fuel system (AFS) directly impacts engine thrust and the safety of the surrounding equipment. To enable timely responses to functional failures within the AFS, this study developed a multi-fault diagnostic method for a simplified AFS using machine learning. Data for this research were sourced from a simulation model and a simplified rig representing the Boeing 777 fuel system. The simulation model enhanced the rig data by introducing random uncertainty, generating sufficient samples across all fault classes. The machine learning candidates were selected from classic algorithms to ensure comparable complexity, predictive capability, and varying levels of transparency. By diagnosing the full combinations of faults in the AFS, this study uniquely transformed a typical multi-label problem—traditionally requiring multiple algorithms—into a multiclass problem solvable by a single algorithm. The evaluation of diagnostic accuracy innovatively incorporated the definition and exploration of algorithmic structural complexity, enabling a fairer comparison among the methods. After evaluating predictive performance, both local (XAI for opaque algorithms) and global (interpretation for transparent algorithms) explainable techniques were applied to address the gap in explainability for fuel system diagnostics. This evaluation framework uniquely quantified and compared the interpretability of four ML algorithms to identify the most suitable diagnostic method and tackle the three primary challenges outlined in Section 1.

Given its predictive performance, the ANN emerged as the most suitable algorithm for this multi-fault diagnostic task. However, the interpretability provided by XAI tools did not always align well with engineering insights. While the DT demonstrated the highest interpretability consistency, this came at a cost to the overall performance. DTs also offer the advantage of easy rule extraction in simpler structures. From an engineering perspective, a method that excels at interpretability but falls short in fault detection is limited in its effectiveness, particularly when maintenance decisions depend on its output.

This study emphasises the need for further research into machine learning interpretability to serve domain users better. Engineering knowledge should remain a central consideration in designing and implementing data-driven algorithms, and it plays a vital role in validating XAI outputs.

Author Contributions: Conceptualization, J.L., S.K. and I.J.; methodology, J.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; investigation, J.L.; resources, J.L.; data curation, J.L.; writing-original draft preparation, J.L.; writing—review and editing, S.K. and I.J.; visualization, J.L.; supervision, S.K. and I.J.; project administration, I.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sciatti, F.; Tamburrano, P.; De Palma, P.; Distaso, E.; Amirante, R. Detailed simulations of an aircraft fuel system by means of Simulink. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2385, p. 012033.
2. Gao, Z.; Song, D. Research of aircraft fuel system feeding failure based on flowmaster simulation. In *Proceedings of the First Symposium on Aviation Maintenance and Management-Volume I*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 45–52.
3. Zhao, Y.; Li, Z.; Wang, Z.; Xu, R.; Ding, E. Fault-Tolerant Center of Gravity Control for Fuel Systems with Component Failures. In *Advances in Guidance, Navigation and Control: Proceedings of 2020 International Conference on Guidance, Navigation and Control, ICGNC 2020, Tianjin, China, 23–25 October 2020*; Springer: Singapore, 2022; pp. 4327–4336.
4. Li, J.; King, S.; Jennions, I. Intelligent fault diagnosis of an aircraft fuel system using machine learning—A literature review. *Machines* **2023**, *11*, 481. [CrossRef]
5. Althnian, A.; AlSaeed, D.; Al-Baity, H.; Samha, A.; Dris, A.B.; Alzakari, N.; Abou Elwafa, A.; Kurdi, H. Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Appl. Sci.* **2021**, *11*, 796. [CrossRef]
6. Singh, R.; Maity, A.; Somani, B.; Nataraj, P.S. On-board fault diagnosis of a laboratory mini SR-30 gas turbine engine. *IFAC-PapersOnLine* **2022**, *55*, 153–158. [CrossRef]
7. Matei, I.; Piotrowski, W.; Perez, A.; de Kleer, J.; Tierno, J.; Mungovan, W.; Turnewitsch, V. System resilience through health monitoring and reconfiguration. *ACM Trans. Cyber-Phys. Syst.* **2024**, *8*, 1–27. [CrossRef]
8. Chaabane, A.; Jemmali, M. Gas turbine fault diagnosis based on machine learning techniques. In Proceedings of the 2023 IEEE Afro-Mediterranean Conference on Artificial Intelligence (AMCAI), Hammamet, Tunisia, 13–15 December 2023; pp. 1–6.
9. Miao, Y.; Li, Y.; Pan, J.; Liu, Z.; Liu, L.; Wang, Z.; Wang, Z. Bio-Inspired Fault Diagnosis for Aircraft Fuel Pumps Using a Cloud-Edge System. *Biomimetics* **2023**, *8*, 601. [CrossRef] [PubMed]
10. Bai, M.; Liu, J.; Long, Z.; Luo, J.; Yu, D. A comparative study on class-imbalanced gas turbine fault diagnosis. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2023**, *237*, 672–700. [CrossRef]
11. Nekoonam, A.; Montazeri-Gh, M. Noise-robust gas path fault detection and isolation for a power generation gas turbine based on deep residual compensation extreme learning machine. *Energy Sci. Eng.* **2023**, *11*, 4001–4018. [CrossRef]
12. Irani, F.N.; Soleimani, M.; Yadegar, M.; Meskin, N. Deep transfer learning strategy in intelligent fault diagnosis of gas turbines based on the Koopman operator. *Appl. Energy* **2024**, *365*, 123256. [CrossRef]
13. Li, J.; Ying, Y. A Novel Machine Learning Based Fault Diagnosis Method for All Gas-Path Components of Heavy Duty Gas Turbines with the Aid of Thermodynamic Model. *IEEE Trans. Reliab.* **2024**, *73*, 1805–1818. [CrossRef]
14. Salilew, W.M.; Karim, Z.A.; Lemma, T.A. Investigation of fault detection and isolation accuracy of different Machine learning techniques with different data processing methods for gas turbine. *Alex. Eng. J.* **2022**, *61*, 12635–12651. [CrossRef]
15. Dobranská, L.; Biceková, A.; Babič, F. Classification models comparison from the user's level of interpretability. In Proceedings of the 2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 20–22 November 2023; pp. 255–260.
16. Haddada, K.; Khedher, M.I.; Jemai, O.; Khedher, S.I.; El-Yacoubi, M.A. Assessing the Interpretability of Machine Learning Models in Early Detection of Alzheimer's Disease. In Proceedings of the 2024 16th International Conference on Human System Interaction (HSI), Paris, France, 8–11 July 2024; pp. 1–6.
17. Saraf, A.P.; Chan, K.; Popish, M.; Browder, J.; Schade, J. Explainable artificial intelligence for aviation safety applications. In Proceedings of the AIAA Aviation 2020 Forum, Virtual, 15–19 June 2020; p. 2881.
18. Sutthithatip, S.; Perinpanayagam, S.; Aslam, S.; Wileman, A. Explainable AI in aerospace for enhanced system performance. In Proceedings of the 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 3–7 October 2021; pp. 1–7.

19. Li, Y.; Jia, Z.; Liu, Z.; Shao, H.; Zhao, W.; Liu, Z.; Wang, B. Interpretable intelligent fault diagnosis strategy for fixed-wing UAV elevator fault diagnosis based on improved cross entropy loss. *Meas. Sci. Technol.* **2024**, *35*, 076110. [CrossRef]
20. Cummins, L.; Sommers, A.; Ramezani, S.B.; Mittal, S.; Jabour, J.; Seale, M.; Rahimi, S. Explainable predictive maintenance: A survey of current methods, challenges and opportunities. *IEEE Access* **2024**, *12*, 57574–57602. [CrossRef]
21. Brito, L.C.; Susto, G.A.; Brito, J.N.; Duarte, M.A. Fault Diagnosis using eXplainable AI: A transfer learning-based approach for rotating machinery exploiting augmented synthetic data. *Expert Syst. Appl.* **2023**, *232*, 120860. [CrossRef]
22. Sharma, J.; Lal Mittal, M.; Soni, G.; Keprate, A. Explainable Artificial Intelligence (XAI) Approaches in Predictive Maintenance: A Review. *Recent Pat. Eng.* **2024**, *18*, 18–26. [CrossRef]
23. Kim, S.; Choo, S.; Park, D.; Park, H.; Nam, C.S.; Jung, J.Y.; Lee, S. Designing an XAI interface for BCI experts: A contextual design for pragmatic explanation interface based on domain knowledge in a specific context. *Int. J. Hum. Comput. Stud.* **2023**, *174*, 103009. [CrossRef]
24. Aysel, H.I.; Cai, X.; Prugel-Bennett, A. Multilevel Explainable Artificial Intelligence: Visual and Linguistic Bonded Explanations. *IEEE Trans. Artif. Intell.* **2023**, *5*, 2055–2066. [CrossRef]
25. Tao, D.; Song, K.; Xie, H. Research on Fault Diagnosis Method for Diesel Engine Fuel System based on Model-Softmax. In Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 28–30 October 2022; pp. 1–6.
26. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
27. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
28. Breiman, L. *Classification and Regression Trees*; Routledge: London, UK, 2017.
29. Gheraibia, Y.; Kabir, S.; Aslansefat, K.; Sorokos, I.; Papadopoulos, Y. Safety+ AI: A novel approach to update safety models using artificial intelligence. *IEEE Access* **2019**, *7*, 135855–135869. [CrossRef]
30. Kilic, U.; Yalin, G.; Cam, O. Digital twin for Electronic Centralized Aircraft Monitoring by machine learning algorithms. *Energy* **2023**, *283*, 129118. [CrossRef]
31. Vianna, W.O.; Gomes, J.P.; Galvão, R.K.; Yoneyama, T.; Matsuura, J.P. Health monitoring of an auxiliary power unit using a classification tree. In Proceedings of the Annual Conference of the PHM Society, Montreal, QC, Canada, 25–29 September 2011; Volume 3.
32. Giordano, D.; Pastor, E.; Giobergia, F.; Cerquitelli, T.; Baralis, E.; Mellia, M.; Neri, A.; Tricarico, D. Dissecting a data-driven prognostic pipeline: A powertrain use case. *Expert Syst. Appl.* **2021**, *180*, 115109. [CrossRef]
33. Xu, T. A Novel Fault Identifying Method with Supervised Classification and Unsupervised Clustering. *J. Digit. Inf. Manag.* **2013**, *11*, 184–189.
34. Jiang, Y.; Miao, Y.; Qiu, Z.; Wang, Z.; Pan, J.; Yang, C. Intermittent fault detection and diagnosis for aircraft fuel system based on SVM. In *IET Conference Proceedings CP776*; The Institution of Engineering and Technology: Stevenage, UK, 2020; Volume 2020, pp. 1255–1258.
35. Menga, N.; Mothakani, A.; De Giorgi, M.G.; Przysowa, R.; Ficarella, A. Extreme learning machine-based diagnostics for component degradation in a microturbine. *Energies* **2022**, *15*, 7304. [CrossRef]
36. Andrianantara, R.P.; Ghazi, G.; Botez, R.M. Aircraft engine performance model identification using artificial neural networks. In Proceedings of the AIAA Propulsion and Energy 2021 Forum, Virtual, 9–11 August 2021; p. 3247.
37. Yildirim, M.T.; Kurt, B. Engine health monitoring in an aircraft by using Levenberg-Marquardt feedforward neural network and radial basis function network. In Proceedings of the 2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sinaia, Romania, 2–5 August 2016; pp. 1–5.
38. Yildirim, M.T.; Kurt, B. Aircraft gas turbine engine health monitoring system by real flight data. *Int. J. Aerosp. Eng.* **2018**, *2018*, 9570873. [CrossRef]
39. Yao, C.; Yueyun, X.; Jinwei, C.; Huisheng, Z. A novel gas path fault diagnostic model for gas turbine based on explainable convolutional neural network with LIME method. In *Turbo Expo: Power for Land, Sea, and Air*; American Society of Mechanical Engineers: New York, NY, USA, 2021; Volume 84966, p. V004T05A008.
40. Liu, J.; Zhang, Q.; Macián-Juan, R. Enhancing interpretability in neural networks for nuclear power plant fault diagnosis: A comprehensive analysis and improvement approach. *Prog. Nucl. Energy* **2024**, *174*, 105287. [CrossRef]
41. Chen, L.; Li, G.; Liu, J.; Liu, L.; Zhang, C.; Gao, J.; Xu, C.; Fang, X.; Yao, Z. Fault diagnosis for cross-building energy systems based on transfer learning and model interpretation. *J. Build. Eng.* **2024**, *91*, 109424. [CrossRef]
42. Gawde, S.; Patil, S.; Kumar, S.; Kamat, P.; Kotecha, K. An explainable predictive maintenance strategy for multi-fault diagnosis of rotating machines using multi-sensor data fusion. *Decis. Anal. J.* **2024**, *10*, 100425. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.