



COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR HEAT EXCHANGERS DIAGNOSIS IN ELECTRIFIED AIRCRAFT

D. F. Migliore¹, G. D'Alessio², S. Caggese², A. De Martin², F. Acerra³, M. Sorli² & M. Fioriti²

¹Leonardo Labs - Future Aircraft Technologies, Turin (TO), Italy

²Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Turin (TO), Italy

³Leonardo Aircraft Division, Pomigliano D'Arco (NA), Italy

Abstract

Optimizing performance in aircraft heat exchangers is crucial, especially for future electrified aircraft where effectively managing the heat produced by high-power electronics stands out as one of the most significant challenges. This study proposes leveraging advanced diagnostic techniques based on Artificial Intelligence (AI) algorithms to enhance efficiency and safety in heat exchangers. These techniques are proficient in assessing their state of health, thereby enabling the implementation of a Condition-Based Maintenance (CBM) approach. As fouling constitutes a primary failure mode for heat exchangers, this paper presents a comparative analysis of Machine Learning (ML) algorithms aimed at evaluating the severity of this specific failure mechanism. Due to the limited availability of data from flight operations related to the behavior of this device, the data for training and testing the data-driven algorithms was generated using a high-fidelity model of an aircraft's Thermal Management System (TMS), capable of simulating its behavior under both healthy and degraded conditions.

Keywords: Machine Learning, Diagnosis, Heat Exchanger, Fouling, Thermal Management System

1. Introduction

The aviation sector has long been a contributor to greenhouse gas emissions. In recent years it has produced approximately 2% of global energy-related CO₂ emissions, and as global air travel continues to expand, so does its carbon footprint [1]. The climate neutrality target within 2050 compels the transition towards cleaner and more sustainable aviation technologies. Electrification of aircraft, by replacing traditional combustion engines with electric propulsion systems, has emerged as a promising avenue to reduce aviation's environmental footprint. Electric propulsion systems, whether powered by batteries, fuel cells, or hybrid configurations, on one side have the potential to drastically reduce carbon emissions by eliminating or significantly minimizing the use of fossil fuels. On the opposing side, the transition to electric propulsion introduces a unique set of challenges, with one of the most critical being the management of heat generated by the high power electronic devices. The Thermal Management System (TMS) in electrified aircraft plays a crucial role in maintaining the correct operating temperatures of various systems, ensuring safety, and optimizing performance. Challenges include the need to dissipate high power heat, especially during take-off and climb phases, while maintaining aircraft efficiency [2].

A safe and efficient TMS is paramount for the success of electrified aircraft where overheating can compromise the safety of the aircraft, leading to catastrophic failures. To ensure TMS reliability and performance, advanced diagnostic techniques can play key role. While traditional diagnostic methods heavily rely on manual inspection and rule-based approaches, modern strategies progressively exploit sensor data, real-time monitoring, and analytical tools. These approaches can be categorized into three main classes: data-driven, model-based, and hybrid [3]. Data-driven techniques do not require system knowledge but only historical data of the system behaviour. The main advantage of this approach is associated with ease and speed of implementation as well as reduced

computation cost compared to other techniques. While the drawbacks are mainly related to the availability of sufficient amounts of data and to the explainability of the results returned. Model-based or physics-based approaches necessitate understanding both the system and its failure mechanisms. In model-based methods, the behaviour of the healthy and faulty system is described through mathematical equations. This approach allows early fault detection and complete comprehension of the output. Conversely, developing accurate physics-based models require a deep understanding of the system's physics leading to a complex and time-consuming modelling. Hybrid or fusion techniques are based on the combination of the two previous approaches. Hybrid solutions aim to limit both data-driven and physics-based disadvantages overtaking the need of large quantity of historical data and accurate mathematical models [4].

Fault diagnosis in complex systems has undergone a significant evolution with the emergence of data-driven algorithms [5] allowing a comprehensive and proactive approach to identifying faults and abnormalities. By analyzing large volumes of sensor data and operational parameters, data-driven algorithms can detect subtle patterns indicative of impending issues, contributing to early diagnosis. The main objective of this study is to assess the efficacy of different data-driven algorithms, utilizing Machine Learning (ML) techniques, in diagnosing fouling in heat exchangers. Given the scarcity of flight operation data concerning the behavior of heat exchangers in both healthy and degraded conditions, synthetic data was generated using a high-fidelity model of the TMS implemented in Simcenter Amesim software.

The remainder of this paper is structured as follows: section 2 elucidates the functioning of the aircraft's thermal management system, explores the fouling phenomena in the heat exchanger, and details the methodology for generating and analyzing synthetic data. In section 3, we conduct a thorough comparison of the performance of the algorithms under examination. Finally, section 4 presents the conclusions drawn from our study.

2. Thermal Management System

The transition towards electric propulsion systems presents significant challenges concerning the effective management of heat generated by high-power electronics. The establishment of a safe and efficient thermal management system is of utmost importance in ensuring optimal performance and reliability. Integrating advanced diagnostic techniques emerges as a promising approach to address these challenges. Figure 1 presents a simplified schematic of a TMS utilized in aviation applications, with the primary components detailed in the following.

- **Electric pump:** Mainly comprising an inverter, an electric motor, and a pump, it is responsible for the circulation of coolant through the system, enabling the fluid to transport heat from the heat source to the heat exchanger.
- **Heat source:** The device or set of devices that generate heat and require temperature regulation.
- **Expansion tank with pressure cap:** Designed to open when the pressure exceeds a target limit, it serves as safety and regulating device, maintaining the target pressure and reducing the risk of overheating.
- **Overflow tank:** A container that accommodate the expansion of coolant fluid when the system heats up.
- **Heat exchanger:** Design to transfer heat from the hot coolant to the ram air.
- **Piping:** The network of conduits that connect all the components in the TMS, allowing the coolant flow between them.

To assess the reliability of the thermal management system and its constituent elements, an exhaustive Fault Tree Analysis (FTA) was conducted. The highest level of the FTA, illustrated in Figure 2, highlights that the electric pump and heat exchanger exhibit the highest fault ratios among the system components under consideration. Given the relative ease of implementing redundancy in the electric pump, the focus has been directed toward mitigating potential vulnerabilities in the heat exchanger.

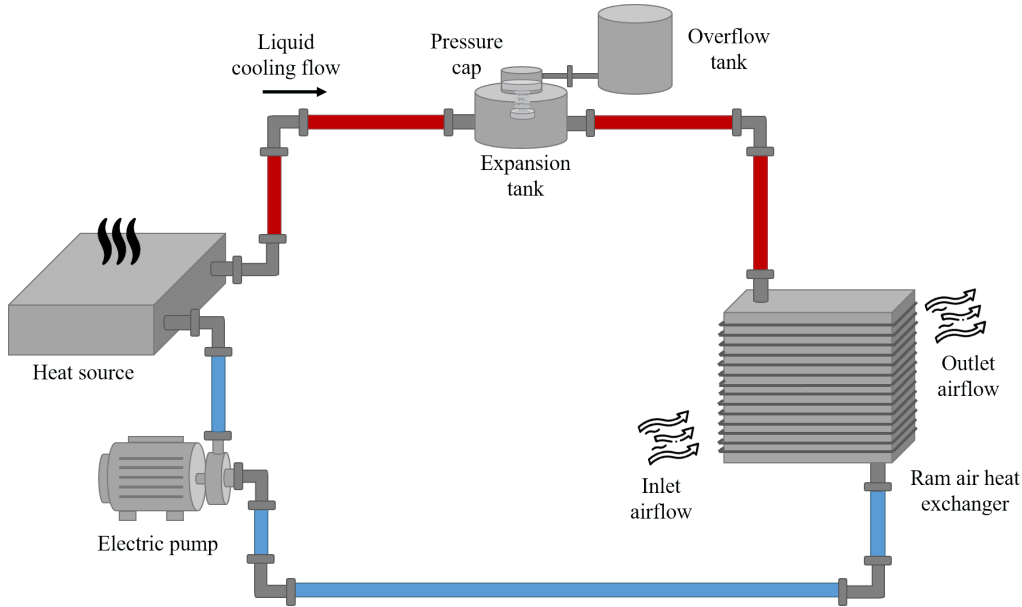


Figure 1 – Thermal management system scheme.

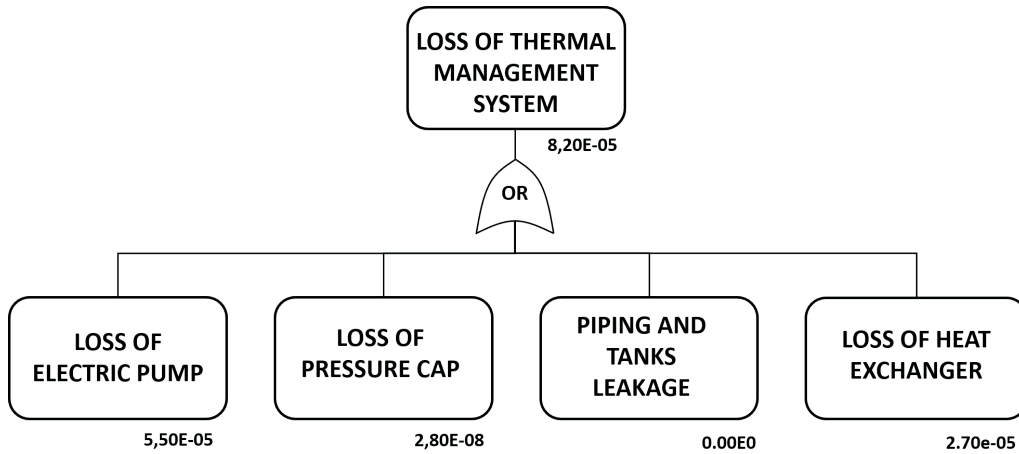


Figure 2 – First level of the TMS fault tree analysis.

2.1 Thermal management system model

The TMS model was implemented utilizing Simcenter Amesim software, facilitating the development of multi-domain dynamic models with pre-defined components libraries. Key libraries employed in model construction include: *Electric Motors and Drives* for the electric drive system, *Thermal Hydraulic* for various components such as the pump, expansion tank, pressure cap, overflow tank, and pipes, *Thermal* for the heat source, *Heat* for the heat exchanger, and *Aerospace and Marine* for modeling aircraft flight missions.

The model comprises several main components. The electric drive system model encompasses a BrushLess Direct Current (BLDC) motor controlled with a torque command signal, an inverter connected to a high-voltage DC-link, sensors, and a control unit. Notably, the efficiencies of the motor and inverter, as well as maximum torque, power, and speed, are assumed constant. Mechanical power delivered by the e-drive is computed by multiplying electrical power at the input of the inverter by global efficiencies of the inverter and motor. Mechanical power is then transmitted to a fixed displacement ideal pump, where flow rate is determined solely by shaft speed and inlet pressure, neglecting flow leakage and mechanical losses. The coolant, whose properties vary with temperature, proceeds to the heat source, modeled as a thermal mass, where the temperature dynamics is computed based in response to incoming heat fluxes. The pressure cap allows connection between the expansion tank and overflow tank to accommodate fluid expansion due to temperature increases.

Specifically, the expansion tank is modeled as an adiabatic fixed capacity, the pressure cap as a symmetric variable orifice adjusting orifice size based on pressure values, and the overflow tank as an accumulator housing both coolant and pressurized air. Here heat exchange between the liquid and gas phases is not considered. The heat exchanger, a flat plate finned-tube type, enables heat transfer between the coolant and the ram air. The calculation model for exchanged thermal flux is based on the epsilon-NTU method, considering a cross-flow architecture. Additionally, akin to the circuit pipes, the pressure losses in the exchanger are also taken into account using the well-established Darcy-Weisbach model [6]. Finally, the pipes model enables the consideration of both distributed and concentrated pressure losses, predominantly attributed to factors such as relative roughness, Reynolds number, and the layout of the cooling circuit, while disregarding heat exchange with the surroundings.

The model was developed with an emphasis on minimizing computational costs, enabling the execution of a large number of simulations essential for data-driven algorithms. This strategy often bypasses the need for intricate models of components deemed insignificantly contributory or with limited impact on the assessment of fouling severity. Hence, a balance between accuracy and computational costs was pursued, with simplifications applied to certain components wherever feasible.

2.2 Heat exchanger fouling model

After carefully selecting the system's components and setting their parameters, our focus turned to modelling the effects of fouling in the system. As reported in [7], failure mechanisms in heat exchangers culminate in two primary modes: leakage and overheating. While fluid leakages in aircraft applications are typically verified through visual inspections or specialized sensors [8], diagnosing overheating is generally more complex due to the absence of external indicators, primarily resulting from internal degradation mechanisms. In this paper, fouling is considered as a cause of overheating, recognized as one of the primary failure modes in heat exchangers [7]. Fouling refers to the undesired accumulation of insoluble particles on the inner heat transfer surface, resulting in the formation of a deposit layer and consequently reducing system efficiency and performance. These deposits may occur as a result of various phenomena, including precipitation, deposition of particulates, accumulation of chemical reaction byproducts, formation of corrosion product layers, and biological fouling. The presence of a deposit layer primarily manifests in several effects, including the introduction of additional heat transfer resistance, slight enhancement in convective heat transfer, reduction of cross-sectional area in flow passages, and increase in internal pipe roughness.

The analysis of heat exchange is commonly approached through an electrical analogy, wherein each convective and conductive heat transfer term represents a thermal resistance to heat flux. When internal fouling occurs, an additional conductive thermal resistance is introduced compared to pristine operational conditions (Figure 3). Consequently, the overall thermal resistance (R_{tot}) is defined by Equation 1 [9], being it the reciprocal of the product between the overall heat exchange coefficient and the exchange area. This resistance encompasses the internal convective term, the fouling conductive term, the wall conductive term, and the external convective term, respectively.

$$R_{tot} = \frac{1}{h_i A_i} + \frac{k_f}{A_f} + \frac{\ln \frac{D_o}{D_f}}{2\pi\lambda L} + \frac{1}{h_o A_o} \quad (1)$$

Here, h_i and h_o denote the convective heat transfer coefficients inside and outside the tube, respectively, while A_i , A_f , and A_o represent the internal, fouling, and external heat exchange areas. The fouling factor, denoted by k_f , is calculated as the ratio of the thickness of the layer to its thermal conductivity. D_o and D_f stand for the outer and inner tube diameters, and λ and L denote the thermal conductivity coefficient and the length of the tube, respectively.

This phenomena leads to higher outlet temperatures of the refrigerant fluid, potentially resulting in overheating issues. Conversely, fouling-induced reduction in passage area necessitates higher velocities to maintain flow rates, these elevated velocities intensify the rate of convective heat transfer between coolant and fouled surface. However, this enhancement is insufficient to compensate the diminished heat transfer attributed to the increased thermal resistance from fouling. Furthermore, the decrease in cross-sectional area within flow passages resulting from sedimentation of deposits

on the exchange surface influences the pressure drop between the heat exchanger's inlet and outlet. According to the Darcy-Weisbach equation provided below, it becomes evident that the pressure drop is inversely proportional to the fifth power of the hydraulic diameter.

$$\Delta p = \frac{1}{2} \rho v^2 \frac{L}{D_h} f_D \quad (2)$$

Here, Δp represents the drop of pressure between the heat exchanger inlet and outlet, v denotes the fluid velocity, L and D_h stand for the length and hydraulic diameter of the tube, respectively, and f_D denotes the Darcy friction factor, which is equal to four times the Fanning friction factor.

Finally, an elevation in surface roughness commonly results in heightened pressure drops due to escalated turbulence within the fluid dynamics regime. Notably, in the presence of laminar flow, the friction factor is decoupled from surface roughness, being solely contingent upon the Reynolds number.

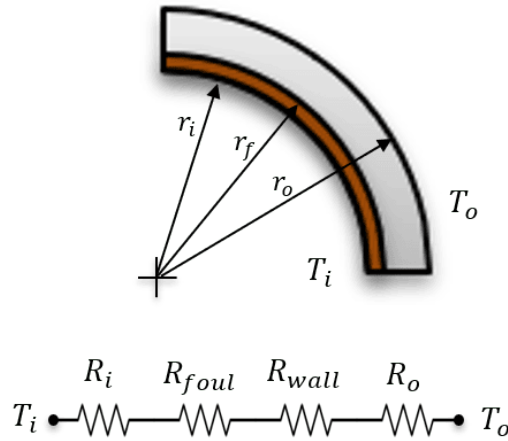


Figure 3 – Pipeline section with the presence of fouling.

2.3 Numerical simulations

Fifteen hundred simulations were conducted to generate the dataset utilized for training and testing machine learning algorithms. Specifically, five hundred simulations were carried out for each degradation level. Following the establishment of a maximum or critical fouling value derived from the temperature requirements of the heat source, the fouling classes are indicated as *no fouling*, *moderate fouling*, and *severe fouling*. The first corresponds to the nominal condition of the heat exchanger and the absence of contamination, ensuring optimal heat transfer. The second represents an intermediate level between the optimal and faulty conditions, serving as an alert for maintenance. The third is considered a critical level, as surpassing this threshold results in temperatures exceeding those specified in the heat source requirements. In particular, the learning models employed in this study were designed with the aim of creating multi-class classifiers capable of recognizing the level of fouling corresponding to a specific flight mission, based on modeled fault levels. The decision to divide fouling into discrete levels and the consequent use of classifiers instead of regressors is based on the concept that, since the temporal evolution of the defect is slow, the fault levels can be thought of as thresholds that serve as warnings to plan maintenance strategies. From this perspective, a division into levels of fault evolution is sufficient, and having a precise estimate of the extent of fouling is not essential, as the focus is on monitoring thresholds rather than continuous values. Hence, owing to its simplicity and effectiveness, classification emerged as the preferred approach.

To encapsulate the inherent variability within mission profiles, static air temperature, static air pressure, true air speed, and heat power dissipation values during the cruise phase were randomly chosen in each simulation. These selections were sampled from Gaussian distributions tailored to the reference class of the aircraft in question. The distributions portraying these parameters are depicted in Figure 4. Within this figure, the after mentioned quantities were normalized by dividing them by

their mean values for industrial considerations. In constructing the synthetic dataset, a deliberate choice was made to concentrate on a specific point within the mission profile where thermal steadiness of the coolant was achieved, rather than analyzing time-series data for the entire mission profile. By concentrating on this specific operational point, the dataset was designed to encapsulate critical insights and characteristics representative of the system's behavior during a stable thermal state. This approach to dataset creation ensures that the ML algorithms are trained and tested on a specific operational condition, contributing to the reliability and efficacy of the subsequent analyses and model outcomes.

A Python script was utilized to size the TMS, and its output was later integrated into the dynamic model. This model incorporates inputs from another Python script, which is responsible for defining simulation parameters such as fouling levels, environmental conditions, and mission profiles. The data generated by the Amesim model is subsequently analyzed and used for both training and testing machine learning algorithms under investigation. The sequential steps of this workflow are visually depicted in Figure 5.

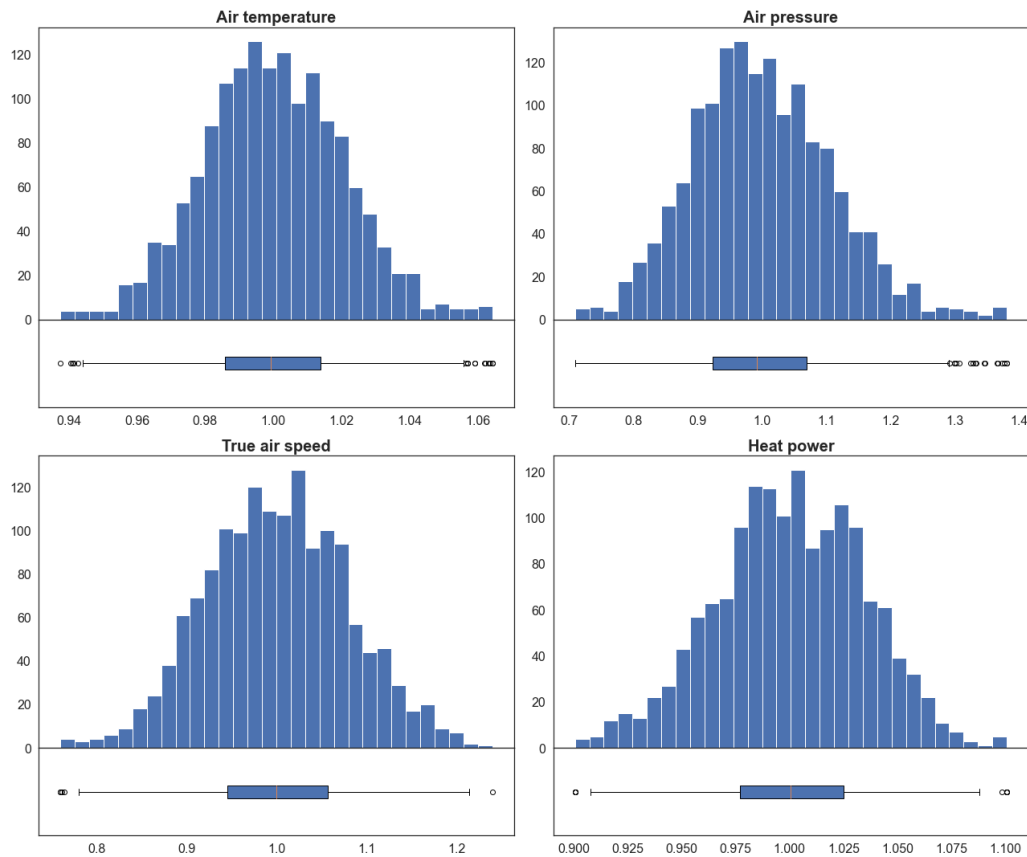


Figure 4 – Workflow illustration depicting the sequential steps involved in sizing and modeling the TMS using Python scripts and Simcenter Amesim software.

2.4 Exploratory data analysis

In this study, the dataset has been generated to achieve a balanced representation across three distinct fouling classes. Through a synthetic generation process, we ensure data cleanliness, eliminating missing values, duplicate entries, and structural anomalies aiding data analysis and inference regarding real-world datasets by mitigating potential confounding factors.

Figure 6 presents an illustrative example of the trends in inlet heat exchanger temperatures throughout an entire flight mission. The plot displays distinct temperature profiles for each fouling level considered, alongside the temperature threshold dictated by the heat source requirements. To ensure comparability across trends, the mean values of the distributions of input model parameters outlined in the preceding section, such as air temperature, air pressure, true air speed, and heat power, were utilized in each of the three simulations. To maintain industrial confidentiality, all quantities depicted in

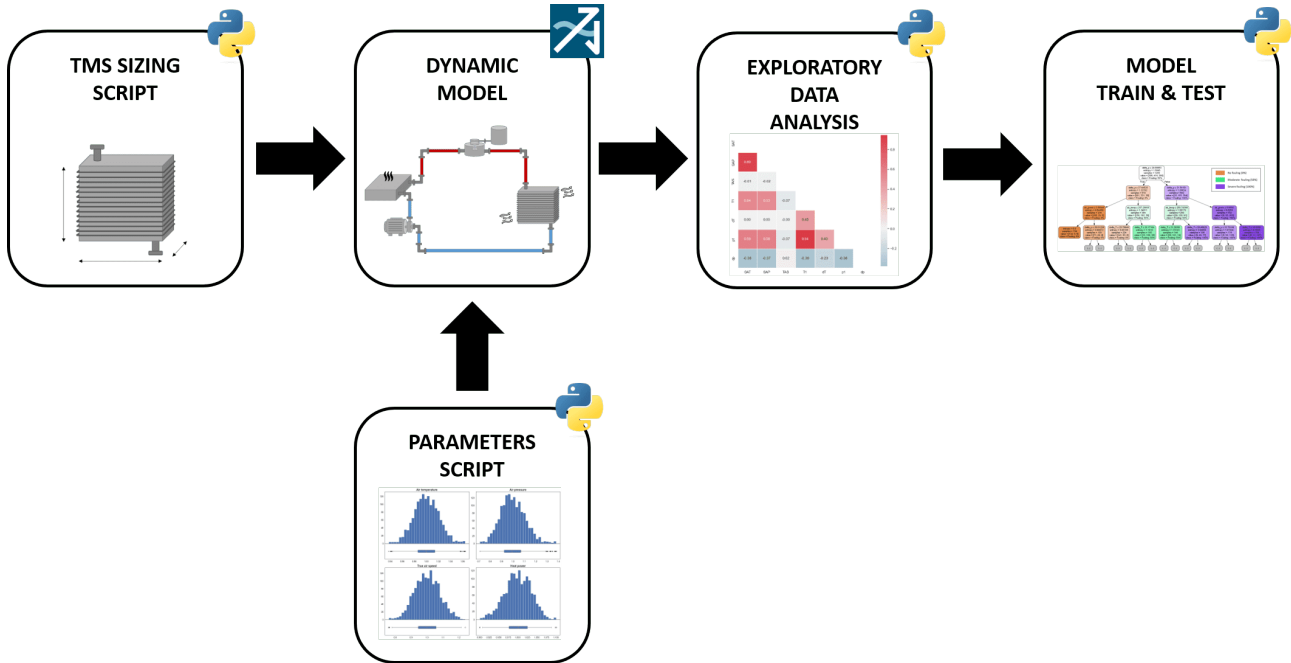


Figure 5 – Workflow from the sizing of the TMS to the training and testing of the ML algorithms.

the figure have been appropriately scaled. Specifically, temperature values are scaled by the threshold value, altitude values are scaled by the cruise phase altitude, and mission time is expressed as a percentage. Figure 7 complements our analysis by showcasing boxplots that depict the distribution of inlet and outlet temperatures and pressures of the heat exchanger across the fouling levels considered. For the same reason reported previously, both temperature and pressure values have been scaled by their respective mean values. Notably, as fouling severity increases, higher values are observed for each considered quantity. Furthermore, Figure 8 provides insights into the interdependencies within the dataset through a correlation matrix analysis, leveraging the Pearson correlation coefficient. This coefficient quantifies the strength and direction of linear relationships between variable pairs, offering valuable insights into dataset dynamics. Key features analyzed include Static Air Temperature (SAT), Static Air Pressure (SAP), True Air Speed (TAS), coolant temperature at the heat exchanger inlet (T1), coolant temperature difference between heat exchanger inlet and outlet (dT), coolant pressure at the heat exchanger inlet (p1), and coolant pressure difference between heat exchanger inlet and outlet (dp). Significant correlations emerge between variables such as static air temperature and static air pressure, as well as between the temperature and pressure at the heat exchanger inlet. Conversely, weaker correlations are observed between environmental parameters (SAT and SAP) and true air speed, as expected.

3. Data driven algorithms comparison

As shown in Figure 9, machine learning algorithms are mainly classified into four categories: (i) supervised learning, (ii) unsupervised learning, (iii) semi-supervised learning, and (iv) reinforcement learning [10, 11]. In the following these techniques are briefly described.

- (i) *Supervised learning* techniques are employed in scenarios involving labeled datasets, where the target variable is explicitly known, enabling the resolution of both classification and regression problems. Classification involves the assignment of data points to distinct classes, while regression entails the estimation of continuous target variables [12].
- (ii) *Unsupervised learning* techniques are utilized in situations where labeled data is absent or when the objective is to uncover underlying structures and patterns within a dataset. Unlike supervised methods, unsupervised techniques operate without explicit target variables, instead emphasizing the intrinsic organization of data. Key tasks in unsupervised learning encompass clustering, anomaly detection, feature learning, and more [10].

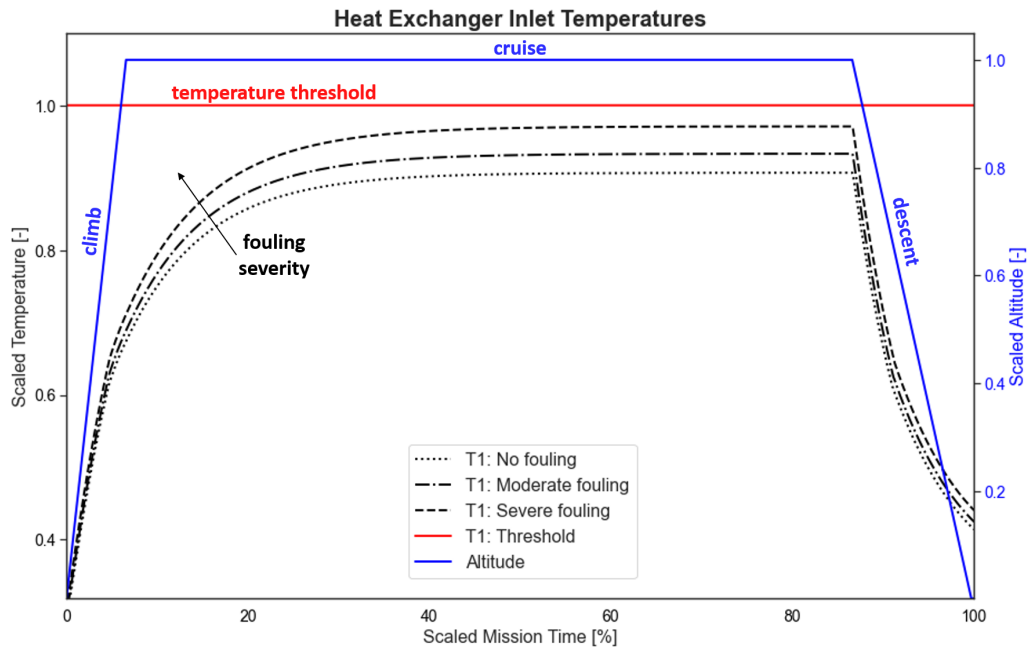


Figure 6 – Inlet heat exchanger temperature trends across varied fouling levels.

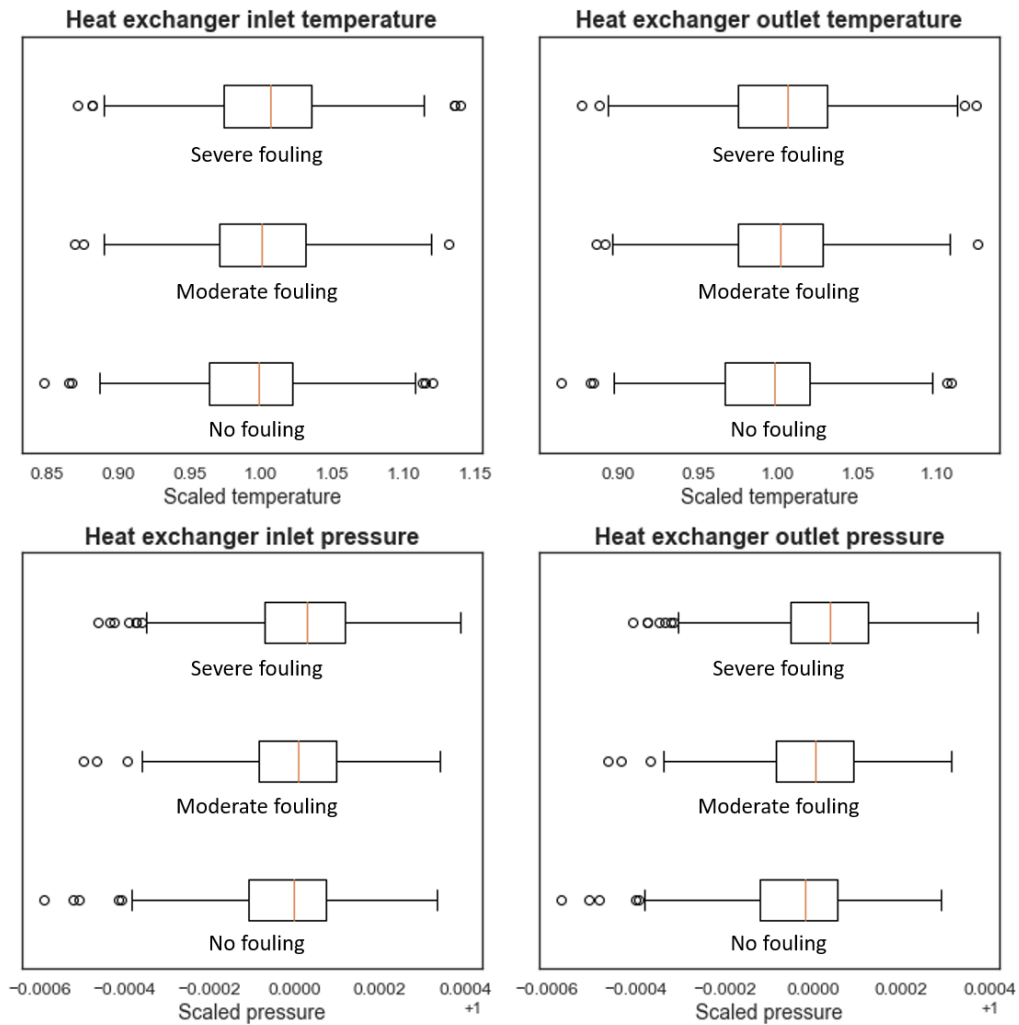


Figure 7 – Inlet and outlet heat exchanger temperatures and pressures.

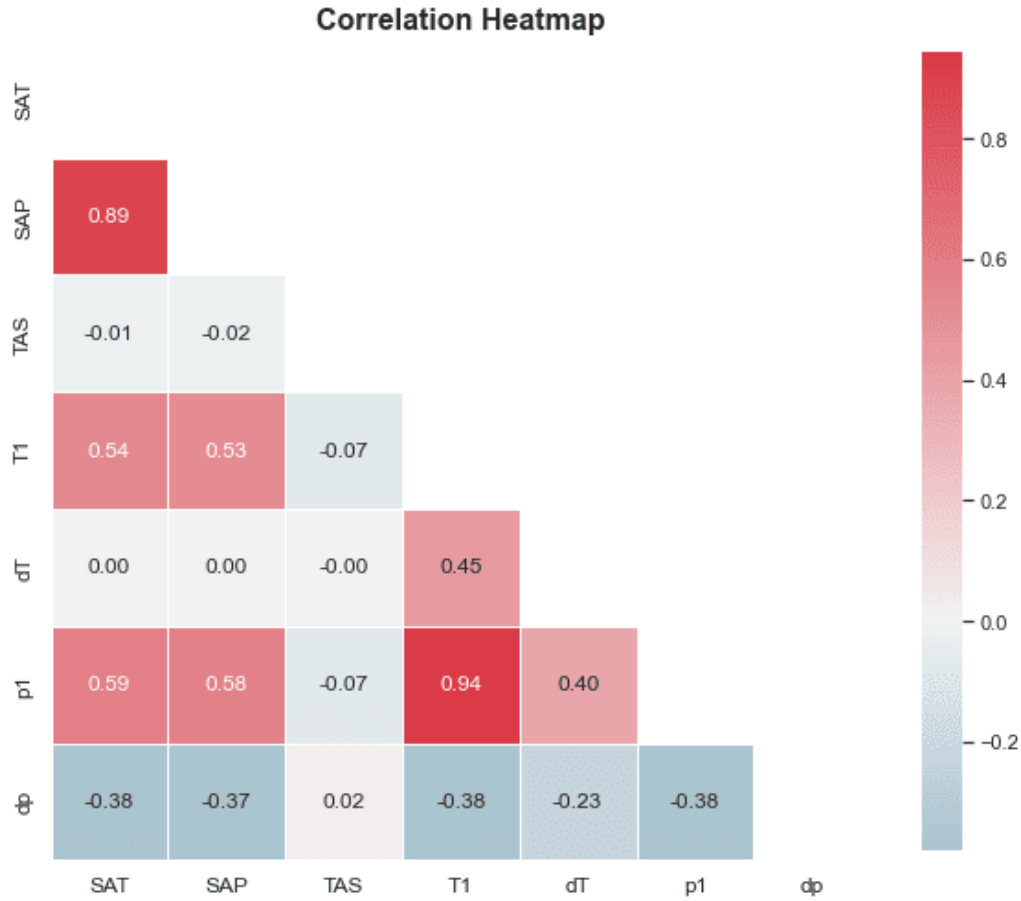


Figure 8 – Features correlations matrix heatmap.

- (iii) *Semi-supervised learning* techniques are employed in scenarios where datasets comprise both labeled and unlabeled data points, combining elements from both supervised and unsupervised methods. The objective of a semi-supervised learning model is to enhance predictive performance beyond what can be achieved by exclusively relying on the labelled data [11].
- (iv) *Reinforcement learning* techniques are based on reward and penalty, allowing software agents and machines to automatically evaluate the optimal behaviour to improve their performance [13].

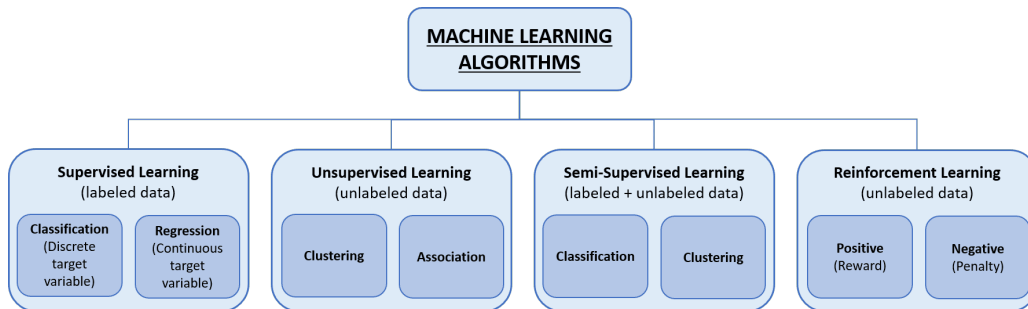


Figure 9 – Machine learning algorithms classification.

The following section presents a comprehensive assessment of the performance of six classification algorithms within the context of a supervised learning problem. Classification problems are typically divided into three distinct categories. The first, known as *binary classification*, pertains to scenarios involving only two classes. The second, termed *multiclass classification*, is applicable when there are more than two classes. The last category, referred to as *multi-label classification*, arises when a single instance may be associated with multiple labels. In this study, multiclass classification methods

are utilized to assess the severity of heat exchanger fouling, classifying it into three levels: *no fouling*, *moderate fouling*, and *severe fouling*.

3.1 Naive Bayes

The Naive Bayes (NB) algorithm is a probabilistic classification method that relies on the assumption of conditional independence among features given the class label, facilitating efficient computations. The algorithm calculates class probabilities using Bayes' theorem, combining prior probabilities and likelihoods. Its simplicity, computational efficiency, and respectable predictive performance make it a valuable tool in the realm of probabilistic classification [14]. The Python *scikit-learn* library provides several Naive Bayes algorithms, offering a range of options. For our dataset, the Gaussian Naive Bayes method was deemed most suitable. Due to its simplicity, default parameter values were adopted, yielding training and testing accuracies of 70.83% and 65.00%, respectively. A comprehensive assessment of model performance is presented in Table 1, showcasing metrics such as precision, recall, and F1-score. Additionally, Figure 10(a) presents the confusion matrix, offering a visual depiction of classification results and insights into the model's performance across different classes. Figure 11(a) displays the Receiver Operating Characteristic (ROC) curve, indicating the performance of the classification by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at varying threshold values. ROC curves are commonly applied in binary classification tasks, where TPR and FPR are clearly defined. In multi-class classification scenarios, TPR or FPR can be derived by converting the output into binary form. Here, a one-vs-rest strategy is employed, comparing each class against the remainder treated as a single entity.

Table 1 – Classification report for Naive Bayes algorithm.

	precision	recall	F1-score	support
<i>no fouling</i>	0.70	0.80	0.75	100
<i>moderate fouling</i>	0.48	0.41	0.44	100
<i>severe fouling</i>	0.74	0.74	0.74	100
accuracy			0.65	300

3.2 K-nearest neighbors

The k-Nearest Neighbors (k-NN) algorithm is a non-parametric and instance-based method grounded in the principle of proximity-based classification. K-NN operates by assigning a class label to an input data point based on the consensus of the classes of its k-nearest neighbors in the feature space [15]. While Naive Bayes models are generally insensitive to feature scaling, k-NN relies on distance metrics, calculating distances between data points for predictions. Thus, it is advisable to normalize or scale features to prevent any single feature from unduly influencing the distance computation. This normalization ensures that all features contribute uniformly to the distance calculation. In the *scikit-learn* library, various scaling techniques are available. For our dataset, we opted for the *StandardScaler* due to its suitability for our use case. In order to enhance the performance of k-NN and subsequent algorithms, it is imperative to tune their hyperparameters. In this study, we conduct hyperparameter tuning using the *GridSearchCV* function available in the *scikit-learn* library. Through a systematic exploration of hyperparameter combinations, *GridSearchCV* facilitates the identification of optimal parameter values, contributing to enhanced model performance. This rigorous tuning process ensures that the k-NN algorithm is tailored to the specific characteristics of the dataset, promoting robust and effective classification results. Following the tuning of hyperparameters the model achieved an accuracy of 89.75% on the training dataset, showcasing its capability to effectively capture patterns within the training samples. While, on the testing dataset, the model exhibited an accuracy of 85.00%, indicating its ability to generalize well to previously unseen data. The detailed classification report for the k-NN model is provided in Table 2, while Figure 10(b) and Figure 11(b) visually present its confusion matrix and its ROC curves.

Table 2 – Classification report for k-Nearest Neighbors algorithm.

	precision	recall	F1-score	support
<i>no fouling</i>	0.89	0.91	0.90	100
<i>moderate fouling</i>	0.81	0.72	0.76	100
<i>severe fouling</i>	0.84	0.92	0.88	100
accuracy			0.85	300

3.3 Decision tree

Decision Tree (DT) algorithms are used for both classification and regression problems. For classification task, the algorithm operates by partitioning the input space into regions, each assigned to a class label, and makes predictions based on these regions. The partitioning is done by recursively splitting the input space based on feature values, resulting in a tree-like structure [16]. The tree was generated using the *DecisionTreeClassifier* from the *scikit-learn* library, with hyperparameters optimized through *GridSearchCV*. The model exhibited a training accuracy of 88.25% and a test accuracy of 85.00%. The classification report of the DT metrics is reported in Table 3 while in Figure 10(c) and Figure 11 are shown, respectively, its confusion matrix and its ROC curves.

Table 3 – Classification report for decision tree algorithm.

	precision	recall	F1-score	support
<i>no fouling</i>	0.87	0.91	0.89	100
<i>moderate fouling</i>	0.82	0.71	0.76	100
<i>severe fouling</i>	0.86	0.93	0.89	100
accuracy			0.85	300

3.4 Random forest

Random Forest (RF) is an ensemble technique [17] that fits several decision tree in parallel using different subset of the dataset. RF uses majority voting for classification tasks or averages for regression task to get the final result. This method allows to reduce the over-fitting and it is generally more accurate than a single decision tree model. Using the random forest algorithm from the *scikit-learn* library and optimizing its hyperparameter through the function *GridSearchCV*, we obtained a training accuracy of 90.92% and a test accuracy of 88.00%. In Table 4, Figure 10(d), and Figure 11(d) are reported, respectively, the classification report of the RF metrics, the confusion matrix, and the ROC curves.

Table 4 – Classification report for random forest algorithm.

	precision	recall	F1-score	support
<i>no fouling</i>	0.90	0.90	0.90	100
<i>moderate fouling</i>	0.83	0.81	0.82	100
<i>severe fouling</i>	0.91	0.93	0.92	100
accuracy			0.88	300

3.5 Adaptive boosting

Adaptive Boosting (AdaBoost), first introduced by [18], is an ensemble learning technique characterized by an iterative approach aimed at enhancing weak classifiers by learning from their errors.

Unlike random forests, which employ parallel ensembling, AdaBoost utilizes a sequential ensemble approach. By combining multiple underperforming classifiers, AdaBoost constructs a robust classifier that typically achieves higher accuracy. AdaBoost is recognized as an adaptive classifier, significantly improving classifier efficiency. However, there is a risk of overfitting in certain scenarios. Applying the AdaBoost algorithm through the *scikit-learn* library resulted in training and test accuracies of 90.75% and 87.33%, respectively. Additional metric values are provided in Table 5, while the confusion matrix and ROC curves are depicted in Figures 10(e) and 11(e).

Table 5 – Classification report for adaptive boosting algorithm.

	precision	recall	F1-score	support
<i>no fouling</i>	0.90	0.87	0.88	100
<i>moderate fouling</i>	0.81	0.81	0.81	100
<i>severe fouling</i>	0.91	0.94	0.93	100
accuracy			0.87	300

3.6 Support vector machine

The Support Vector Machine (SVM) algorithm serves both classification and regression tasks [19]. In high-dimensional spaces, SVM establishes hyperplanes, positioned to maximize the margin from the nearest training data points within any class, facilitating robust separation. A wider margin generally corresponds to lower generalization error. SVM's efficacy in high-dimensional spaces is influenced by diverse kernel functions such as linear, polynomial, radial basis function (RBF), and sigmoid. However, in noisy datasets with overlapping classes, SVM performance may degrade [10]. The RBF kernel was selected for our specific use case due to its superior flexibility in handling complex decision boundaries compared to other kernels. Following hyperparameter tuning, the model achieved a training accuracy of 94.33% and a test accuracy of 93.33%. Detailed SVM metrics are presented in Table 6, and corresponding confusion matrix (Figure 10(f)) and ROC curves (Figure 11(f)) are provided.

Table 6 – Classification report for support vector machine algorithm.

	precision	recall	F1-score	support
<i>no fouling</i>	0.95	0.94	0.95	100
<i>moderate fouling</i>	0.90	0.90	0.90	100
<i>severe fouling</i>	0.95	0.96	0.96	100
accuracy			0.93	300

COMPARATIVE ANALYSIS OF ML ALGORITHMS FOR HEAT EXCHANGERS DIAGNOSIS

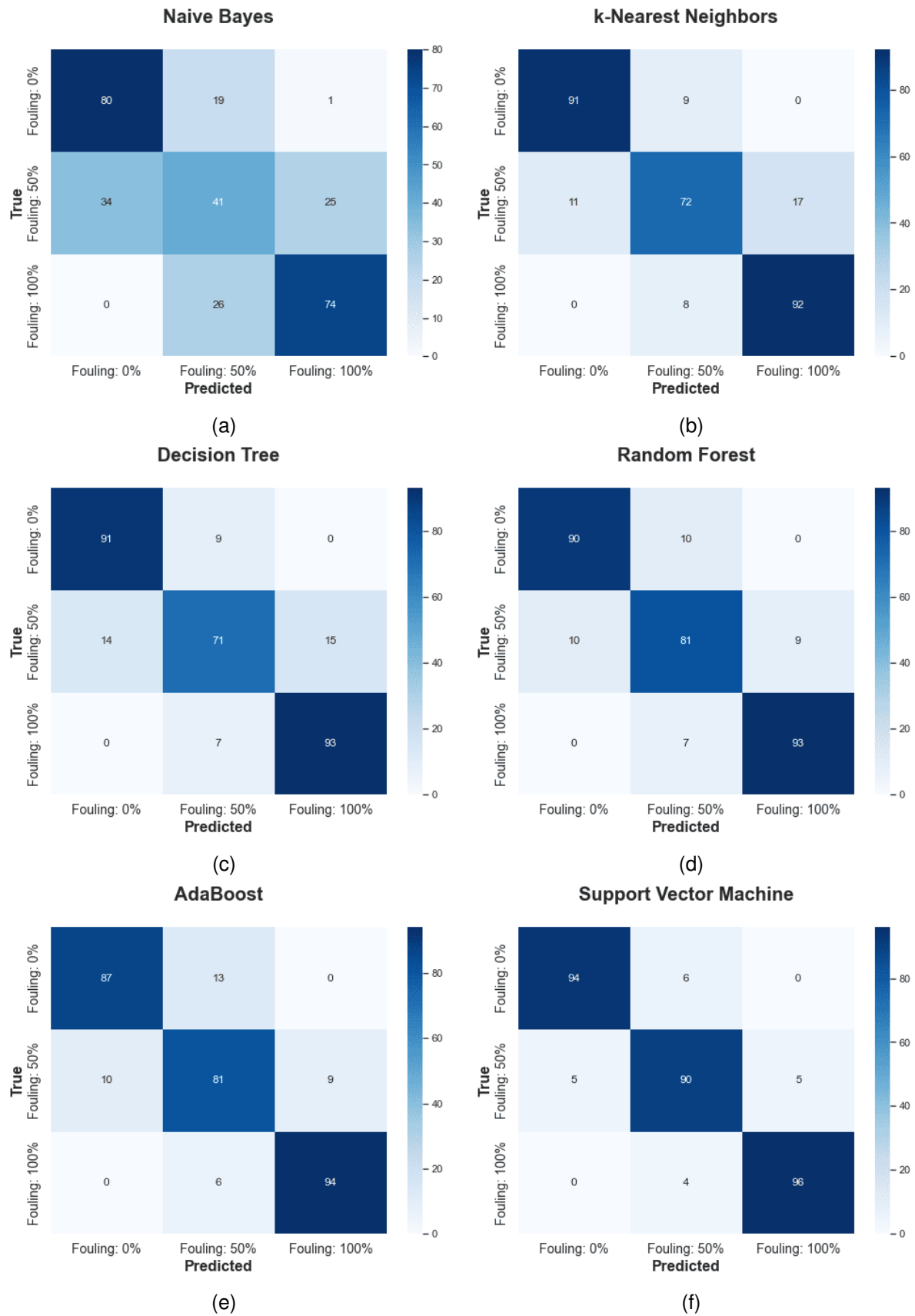


Figure 10 – Confusion matrices for the evaluated algorithms.

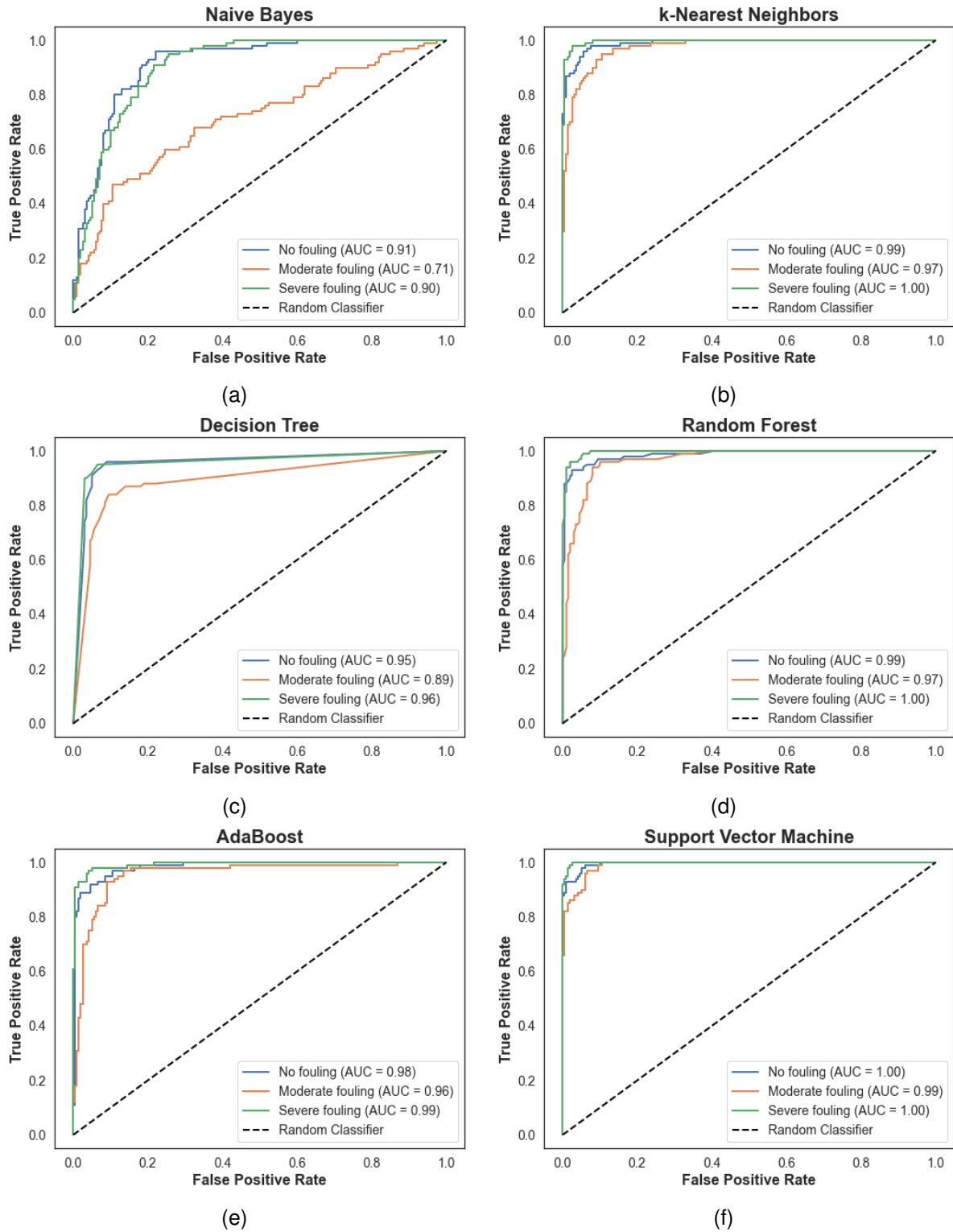


Figure 11 – ROC curves for the evaluated algorithms.

4. Conclusion

In this study, we conducted a comprehensive comparative analysis of six distinct machine learning algorithms — Naive Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, Adaptive Boosting, and Support Vector Machine — applied to assess fouling severity in an aircraft thermal management system's heat exchanger. Through rigorous evaluation, each algorithm's performance and suitability for the task were thoroughly examined. Our findings provide valuable insights into the efficacy and applicability of these algorithms in real-world scenarios. Future research endeavors could explore additional algorithms or refine existing models to further enhance diagnostic accuracy and optimize thermal management systems' efficiency.

5. Acknowledgements

The research presented in this paper has been performed in the framework of the Leonardo Labs in collaboration with the Politecnico di Torino's research units of *Mechatronics and Servosystems* and *Design and Development of Aerospace Systems and Technologies*.

The authors express gratitude to Flavio Di Fede for his valuable assistance in thermal management system modeling and to Manuel Colavincenzo for his expertise in the analysis of machine learning techniques.

6. Contact Author Email Address

mailto: domenico.migliore.ext@leonardo.com

7. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

- [1] 2022 environmental report. Technical report, ICAO, 2023.
- [2] Jeffrey Freeman, Philip Osterkamp, Michael Green, Andrew Gibson, and Benjamin Schiltgen. Challenges and opportunities for electric aircraft thermal management. *Aircraft Engineering and Aerospace Technology: An International Journal*, 86(6):519–524, 2014.
- [3] Khaoula Tidiri, Nizar Chatti, Sylvain Verron, and Teodor Tiplica. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual reviews in control*, 42:63–81, 2016.
- [4] George J Vachtsevanos, Frank Lewis, Michael Roemer, Andrew Hess, and Biqing Wu. *Intelligent fault diagnosis and prognosis for engineering systems*, volume 456. Wiley Hoboken, 2006.
- [5] Yaguo Lei, Bin Yang, Xinwei Jiang, Feng Jia, Naipeng Li, and Asoke K Nandi. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587, 2020.
- [6] Glenn O Brown. The history of the darcy-weisbach equation for pipe flow resistance. In *Environmental and water resources history*, pages 34–43. 2002.
- [7] Murad Ali, Anwar Ul-Hamid, Luai M Alhems, and Aamer Saeed. Review of common failures in heat exchangers—part i: Mechanical and elevated temperature failures. *Engineering Failure Analysis*, 109:104396, 2020.
- [8] Mark Kim and Jeff Demo. Aircraft contaminant and leak detection sensor system for condition based maintenance. In *2017 IEEE Aerospace Conference*, pages 1–9. IEEE, 2017.
- [9] John E Hesselgreaves, Richard Law, and David Reay. *Compact heat exchangers: selection, design and operation*. Butterworth-Heinemann, 2016.
- [10] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160, 2021.
- [11] Mohssen Mohammed, Muhammad Badruddin Khan, and Eihab Bashier Mohammed Bashier. *Machine learning: algorithms and applications*. Crc Press, 2016.

- [12] Iqbal H Sarker, ASM Kayes, Shahriar Badsha, Hamed Alqahtani, Paul Watters, and Alex Ng. Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data*, 7:1–29, 2020.
- [13] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- [14] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. *arXiv preprint arXiv:1302.4964*, 2013.
- [15] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6:37–66, 1991.
- [16] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
- [17] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- [18] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.
- [19] S. Sathiya Keerthi, Shirish Krishnaji Shevade, Chiranjib Bhattacharyya, and Karuturi Radha Krishna Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural computation*, 13(3):637–649, 2001.