Unsupervised Clustering for Internal Combustion Engines Health Monitoring

Bernardo Feijó Junqueira
Rafael Greca Vieira
Bruno Bilhar Karaziack
Ivan de Jesus Pereira Pinto
CPQD

Ivan J. Vianna de Freitas
Previsiown

ABSTRACT

We present an unsupervised machine learning approach designed to aid in the health monitoring of internal combustion engines, enabling the early detection of possible failures or degradation over time. Our methodology uses temporal data collected from a datalogger device connected to a vehicle's On-Board Diagnostics system. This data serves as input for a clustering machine learning model, which is trained incrementally over time to detect anomalies in multivariate time series. The decision-making process to categorize cluster behaviors as anomalies, which might be indicative of engine degradation over time, is based on several key metrics, such as the Jaccard similarity coefficient, relative cluster population, stability, and movement of the centroid over time. The proposed approach is validated on a public rotating machine dataset and tested on an internal combustion engine of a medium-sized vehicle in idle condition.

RESUMO

É apresentada uma abordagem de aprendizado de máquina não supervisionada destinada a auxiliar o monitoramento da saúde de motores de combustão interna, possibilitando a detecção precoce de possíveis falhas ou degradação ao longo do tempo. Nossa metodologia utiliza dados temporais coletados de um dispositivo datalogger conectado ao sistema On-Board Diagnostics de um veículo. Esses dados servem como entrada para um modelo de aprendizado de máquina de agrupamento, treinado incrementalmente ao longo do tempo para detectar anomalias em séries temporais multivariadas. O processo de tomada de decisão para categorizar comportamentos de clusters como anomalias, que podem ser indícios do começo da degradação do motor ao longo do tempo, baseia-se em diversas métricas-chave, como o coeficiente de similaridade de Jaccard, população relativa do cluster, estabilidade e movimento do centróide ao longo do tempo. A abordagem proposta é validada em um conjunto de dados públicos de uma máquina rotativa e testada em um motor de combustão interna de um veículo de tamanho médio em condição de ponto morto.

INTRODUCTION

Health monitoring plays a pivotal role in fault diagnosis and damage prediction in mechanical or dynamical systems [1]. Traditional methodologies for utilizing monitoring data can be classified into model-based and data-driven methods. In the context of combustion engines, the model-based approach usually involves constructing a semi-analytical or a finite-element model of specific engine components, limiting the analysis to individual parts rather than the entire engine, as can be seen in [2-6]. Furthermore, these methods require prior knowledge of the true model to ensure that the estimated parameters fall within an acceptable confidence interval. By measuring discrepancies between model predictions and new measurements, potential damages can be identified [7]. On the other hand, data-driven methods focus on directly analyzing data patterns to detect condition changes, bypassing the need for prior knowledge. Traditional data-driven techniques, which stem from multivariate statistics, approach damage detection as a statistical pattern recognition problem. Despite their widespread application in health monitoring, these methods have limitations. Analyzing all available data is often impractical, and for the sake of computational efficiency, traditional data-driven methods are typically restricted to small datasets [7]. To address these challenges, Machine Learning (ML) approaches are increasingly being utilized, offering enhanced capabilities for handling larger datasets and improving the accuracy of damage predictions [8-12]. Several studies have demonstrated the efficacy of ML in detecting faults and predicting system failures and damages.

Carrara *et al*. [13] propose a Deep Learning (DL) model for time series forecasting to monitor the health of the San Frediano bell tower. By inspecting and detecting anomalies in a large vibration dataset recorded over the long term, they frame the problem as an unsupervised

anomaly detection task. They trained a Temporal Fusion Transformer to learn the structure's normal dynamics using the covariance-driven Stochastic Subspace Identification technique and then used it to detect anomalies by analyzing the differences between the predicted and observed frequencies. The model achieved promising initial results, being able to identify important events, like the Amatrice earthquake and the Santa Croce celebrations, while also being able to detect disturbances caused to the oscillating bells by the religious events that took place in the city's cathedral at weekends. However, the authors pointed out that a comparison between the proposed approach and anomaly detection techniques was not performed.

Junqueira *et al.* [14] proposed an approach that combines a DL model with Gaussian random fields to predict defect field profiles from the interfaces of composite laminates by reading scattered fields from guided modes. The results demonstrated that the developed model is robust against noisy data, interface modeling errors, and reduced models. Additionally, it was tested across various interfaces, directions, and guided modes, consistently achieving high accuracy. This indicates significant potential for application in real-world acoustic inspections in order to monitor the health of composite laminates interfaces while considerably reducing computational time compared to traditional inverse methods.

Xiong *et al*. [15] combine data-driven techniques with a Long Short-Term Memory (LSTM) model to construct a digital twin of an aero-engine for predictive engine maintenance purposes. The results demonstrate the effectiveness of the proposed method, achieving high prediction accuracy and maintaining a low Root Mean Square Error (RMSE) for predicting the Remaining Useful Life (RUL) of the aero-engine. Furthermore, they underscore the importance of exploring unsupervised learning methods for fault diagnosis, with the goal of improving predictive maintenance practices and ensuring the safety of civil aircraft operations.

Jan *et al*. [16] proposed a distributed sensor-fault detection and diagnosis system based on machine learning algorithms. A fault detection block is implemented directly in the sensor to provide immediate output after data collection. This block comprises an auto-encoder that transforms the input signal into a lower-dimensional feature vector, which is then fed into a Support Vector Machine (SVM) for classification as normal or faulty. Subsequently, fault diagnosis is conducted at a central node, such as a network server, to alleviate the computational burden on the sensor. In this study, a Fuzzy Deep Neural Network (FDNN) is employed for diagnosis to offer additional information, such as the type of fault. The input data propagates through a deep neural network and undergoes a fuzzy representation process. The outputs of these components are then fused through densely connected layers. To evaluate the proposed model's performance, data obtained from a healthy temperature-to-voltage converter is utilized, considering five different types of faults: drift, bias, precision degradation, spike, and stuck faults. The results indicate that the proposed model has the efficacy of a fuzzy learning-based model compared to classic neuro-fuzzy and non-fuzzy learning approaches. However, variety in the formats, high levels of uncertainty, noise, and types of data may affect the performance.

Chukwudi e*t al*. [17] proposed a DL-based ensemble model approach in order to predict the condition of vehicle engines. They evaluated the proposed models by using metrics such as RMSE, Root Mean Square Deviation (RMSD), Mean Absolute Error (MAE), accuracy, confusion matrix, and Area Under the Curve (AUC), achieving good performance for a computer-generated dataset. Despite the results, they emphasized the challenges associated with detecting vehicle failures in advance, citing the complex composition of components and sensors. Importantly, they noted that since the model was not tested in a real-world scenario, the results obtained with computer-generated datasets may differ in practical applications; therefore, further investigations in that sense should be performed.

The utilization of ML techniques in engine systems is experiencing a notable surge, encompassing a wide array of applications. These range from forecasting emissions [18, 19] to intricate tasks such as modeling and control, as well as prediction, classification, diagnosis, and fault monitoring [20-23], which constitute the focal point of the current study. Leveraging ML alongside cloud computing and vehicle-to-infrastructure (V2I) communications promises further enhancements in the performance of Internal Combustion Engines (ICE). For instance, ML facilitates the development of efficient, precise, and real-time peer-to-peer learning methods for monitoring performance, data collection and processing, and drawing insights from a vast network of similar engines. This capability paves the way for highly accurate and adaptive engine models, as well as innovative fault detection methodologies enabled by the abundance of training data [24]. It is important to recognize here that, alongside the abundance of data, there are challenges associated with data annotation and defining operational scenarios. That is, in real-world scenarios, it is not always feasible to gather or simulate data for every conceivable analysis case in order to employ supervised learning algorithms. This challenge is particularly pronounced in the realm of engines, which boast a diverse range of components and a multitude of potential faults that can manifest in various forms during operation. For example, consider the study by Junqueira *et al*. [23], which classified three distinct diesel engine faults using audio signals, albeit restricted to these specific scenarios in addition to a healthy engine.

To address the challenges associated with engine health monitoring and data annotation, this study proposes an Internet of Things (IoT)-based solution incorporating an unsupervised ML method to automatically monitor the health of the vehicle ICE. The solution pipeline begins with data acquisition using a datalogger device, which collects

engine data directly from the On-Board Diagnostics (OBD) port. This data is then communicated to a middleware IoT platform, where the ML model collects and processes it in order to generate evolutionary metrics that are monitored over time. The core concept of the ML algorithm is to group the data into clusters over time and use these evolutionary metrics to detect changes in the formation or composition of these clusters. Such changes could indicate a degradation in the health of the ICE.

The remainder of this paper is organized as follows: the methodology section takes place to explain the incremental unsupervised clustering model and the evolution metrics used to monitor the health of a vehicle's ICE, followed by a database section, a section detailing the experimental setup, the results section, and finally, in the final section, we draw the conclusions.

METHODOLOGY

The present work makes use of multivariate time series data with an ML approach based on the study developed by Landauer *et al.* [25], which designed a dynamic, unsupervised clustering model that is trained gradually over time and tracks the transitions of several successive groups in order to identify anomalies in log data using time series data. The model's performance is evaluated using metrics that were designed to assess the clusters' evolution, such as an overlap metric based on the Jaccard coefficient and metrics related to the evolution of the clusters through time, such as the clusters' stability, their relative population, their centroid's movement, and also the clusters' weights according to their centroid's distance to the average center.

The overlap metric used in [25] is based on the Jaccard Coefficient, a measure of similarity between two data sets [26]. Originally proposed by Greene *et al.* for binary sets [27], this metric was adapted by [25] and is defined as follows:

$$overlap(C,\ C') = \frac{|\,(R_{curr} \cap R'_{prev}) \cup (R_{next} \cap R'_{curr})\,|}{|\,(R'_{curr} \cup R'_{prev}) \cup (R_{next} \cup R'_{curr})\,|}\ , \quad (1)$$

where C and C' represent the old and the updated clustering maps, respectively. That is, C is the cluster model trained with data in state t, while C' is the same model incrementally trained with new data related to the transitioning state Δt, generating the state $t' = t + \Delta t$. Furthermore, R_{curr} and R'_{prev} are the data (state t) used to create the cluster map C, inferred by C and C', respectively. R_{next} and R'_{curr} are the new data (represented by transitioning state Δt) used to incrementally train the cluster map from C to C', inferred by C and C', respectively. In order to clarify the meaning of these variables in the context of the present work, an illustrative sketch is presented in Figure 1.

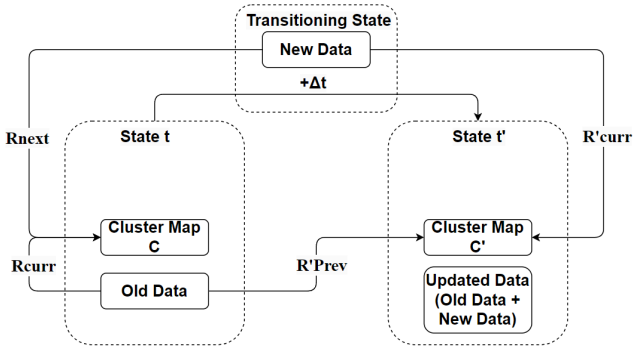

Figure 1. An illustrative sketch of the overlap metric.

The overlap between cluster maps C and C' is a number between $\{0,\ 1\}$, where 1 indicates that all data samples are present in both cluster maps and 0 indicates that both cluster maps are totally different from each other [25]. Here, it is important to highlight a major difference between the overlap metric defined in [25] and the one proposed in the present work. Since we incrementally train our model with a much smaller volume of data relative to the complete volume of data in the current state t, rather than defining a new model at each new state, the clusters do not undergo an abrupt shift between states. That is, a cluster does not fully transform into another between consecutive steps, making it irrelevant to compute overlap metrics between different defined clusters. Therefore, we always consider the metrics computed from cluster C_1 to C'_1 and never from cluster C_1 to C'_2 or C_2 to C'_1, for example (this applies to the subsequent equations as well).

The clusters' stability evolution metric was based on what was proposed by Toyoda and Kitsuregawa [28], which represents the amount of disappeared, merged, appeared, and split data within a specific cluster and was used to measure the evolution of web communities from web archives. We used the relative stability metric elaborated and used by [25], which is defined as follows:

$$stability(C,\ C') = \frac{R'_{prev} + R_{curr} * 2\,|R'_{prev} \cap R_{curr}|}{|\,(R'_{curr} \cup R'_{prev}) \cup (R_{next} \cup R'_{curr})\,|}\ , \quad (2)$$

where the definitions of R'_{prev}, R_{curr}, R'_{curr}, and R_{next} are the same as those defined in Eq. 1, representing the previous and current data samples that were used to create the cluster map at a given state. A high score indicates that only minor changes occurred in the cluster during the state transition, while a low score indicates that major changes took place.

With regards to the clusters' relative population evolution metric, it measures the size of the population (also referred to as the amount of data within the cluster) for each cluster in time step t

$$pop(C) = \frac{L_{Ci,t}}{\sum_{i=1}^{n} L_{Ci,t}} . \quad (3)$$

Here, $L_{Ci,t}$ can be defined as the population of cluster map C_i in time step t, where $i \in n$ (the number of clusters in that time step). This metric helps us to monitor the population clusters through time, making it possible to detect anomalies in their growth behavior (such as if a cluster population starts to grow or drop significantly in a short span of time, it might indicate that the data pattern is changing).

As addressed by [25] and [29], we also track the clusters centroid's movement throughout time because, by tracking their movement, we can understand their behavior, like the clusters' location and direction. Moreover, we assign weights to the clusters according to their centroid's distance relative to the centroids average. The distance is calculated using Eq. 4, where $CE_{Ci,t}$ is the centroid's position of the cluster C_i at time step t and ACE_t is the average between all clusters centroid at that time. Finally, the clusters' weight is determined by Eq. 5, where $W_{Ci,t}$ is the weight of the cluster C_i in the time step t. This approach was developed with the aim of penalizing clusters composed of data considered to be outliers.

$$distance(C) = || CE_{Ci,t} - ACE_t ||, \quad (4)$$

$$weight(C) = \frac{W_{Ci,t}}{\sum_{i=1}^{n} W} , \quad (5)$$

Additionally, we use the Elbow Method to identify the ideal number of clusters to be used and a feature selection algorithm, which selects the features collected with the datalogger by analyzing the correlation between them and, therefore, discarding redundant and unwanted features. Finally, the selected variables are used as input for the ML model under analysis. An overview of the whole methodology is presented in Figure 2.

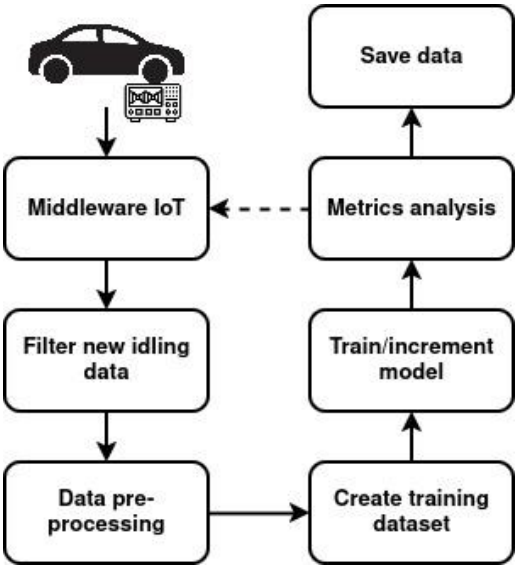

Figure 2. Flowchart of the proposed methodology.

Drawing from Figure 2, the pipeline execution steps are detailed as follows:

- The data is gathered from a datalogger device linked to the vehicle's OBD port.
- Subsequently, this collected data is transmitted to a middleware IoT platform.
- The solution then focuses on collecting only the data related to idling periods from the middleware IoT platform.
- Following this, the filtered data undergoes pre-processing steps.
- The training dataset is gradually constructed until it reaches the necessary volume to either train an initial ML clustering model or increment an existing one.
- Once the model has inferred the data, monitoring metrics are computed. These metrics can be relayed back to the middleware IoT platform for user consultation or to trigger threshold-based alarms.
- Finally, the updated data is stored for further analysis.

DATABASE

In order to validate the proposed method, we used a labeled, public dataset composed of test beds with rotating machines stressed under various conditions, which was created and proposed by Jung *et al.* [30]. The test beds were created by simulating various faults at a sampling rate of 25.6kHz, such as bearing faults, shaft parallel misalignment, and rotor unbalance, under different operating conditions, such as load and constant rotating speed conditions. This dataset has been split into two smaller datasets. The first dataset includes data obtained under various loads and constant rotating speed conditions, while the second dataset includes data obtained under

randomly fluctuating rotating speed conditions without any load. For this work, only the initial dataset is utilized, focusing on the load situation of 4 Nm and exclusively considering unbalance faults. We opted to utilize solely the first dataset because, as demonstrated by the authors of the dataset's publication, it reflects real-world scenarios with load variations and is therefore suitable for evaluating the effectiveness of recently developed models based on rotor dynamics theories.

The data from a medium-sized vehicle was gathered via a datalogger, which is an OBD device that scans the vehicle's dashboard, collects the data, and then sends it to a middleware IoT platform responsible for storing all the data. We obtained a total of 28 different features distributed in numerical and categorical data related to the engine. Since the features had divergent acquisition rates that resulted in NaN values, we applied forward and backward propagation techniques in sequence to fill those values (spreading the last valid observation forward and the next valid observation backward, respectively). After that, we resampled the data according to the timestamp by getting the mean and mode of the numerical and categorical features, respectively, using 10-second frequency windows so that the features had the same acquisition rate. Additionally, the categorical columns were transformed to numerical, and, finally, we scaled the features so that each of them has a zero mean and unit variance. After the data collection and the preprocessing step, we conducted a feature selection evaluation in order to filter the best features and, therefore, reduce the dimensionality of the problem. We ended up with 7 features: engine speed, throttle position, engine torque, engine water temperature, engine load, intake air temperature and engine oil temperature.

EXPERIMENTAL SETUP

In the experimental setup of the vehicle’s ICE, a datalogger device is installed in the vehicle in order to collect data directly from the OBD port. These data are then transmitted to a middleware IoT platform for collection and processing by the ML model. Furthermore, due to the lack of professionals and equipment needed for simulating engine faults, we conducted the experiment on an idling medium-sized vehicle with proper functioning (approximately 800-1000 RPM) consisting of two main steps:

- Step 1: Start the vehicle equipped with the datalogger and leave it in neutral for about 15-20 minutes with the air conditioning off.
- Step 2: Turn the air conditioning on high potency, with the vehicle still in neutral, and leave it for another 15-20 minutes.

This experiment simulating changes to the air conditioning system was designed to get around the inability to simulate engine faults, since when this system is turned on at high power, it makes the engine work harder, resulting in increased heat generation, fuel consumption, and strain on the engine. This, in turn, affects a wide range of vehicle parameters [31]. However, it is important to note that in reality, with a much larger volume of data, data collection should include scenarios with the air conditioning both on and off to cover all possible cases of a vehicle idling, and the considerations made in this experiment are solely to validate the current proposed solution. Moreover, the engine's operating condition is tied to idling, aiming to minimize variations in its operation. For instance, let us hypothetically consider data being collected from a vehicle traversing a large city. If this vehicle embarks on a journey through mountainous terrain, the developed models could mistakenly indicate an engine malfunction, when in reality, the engine is simply operating under different conditions.

For both datasets, the experimental stage configuration is the same. Upon completion of the pre-processing phase, the medium-sized ICE vehicle's data set is progressively sorted based on its timestamp and divided into two smaller sets: one for driving with the air conditioning turned on and the other for driving with it turned off. The machine learning model is then fed incrementally with the two sets after they have been divided into smaller 50 ms chunks. Since this is a model for identifying anomalies, it is crucial to remember that the data from the engine's typical state should be used first (in this case, when the air conditioning is turned off). This will enable the model to look for patterns in the engine's proper operation and spot anomalies that point to engine degradation. The public dataset on machine rotation faults operates on the same premise. In this case, the data that represents the typical rotating machine state is the one that is not faulty. Moreover, as mentioned in the Methodology section, we used the results obtained through the Elbow Method to establish the number of clusters for each dataset.

RESULTS

The results are presented for the two aforementioned datasets. The first one consists of a controlled experimental environment of a rotating machine with unbalanced rotors. Although not directly related to ICEs, it is used to validate the proposed method, which could be generalized to work with a wide range of dynamical systems. The second dataset pertains to the data collected from a real ICE of a medium-sized vehicle with simulated faults, used to test the proposed approach in a real-world scenario. Furthermore, all the presented results utilize a ML model based on mini-batch K-means, chosen for its simplicity in incremental training. It is worth noting that other clustering algorithms with incremental training capabilities could also be employed for this monitoring task.

ROTATING MACHINE DATASET - In Figure 3, the Elbow method is depicted to determine the number of clusters to be defined by the K-means algorithm for the rotating machine dataset. The Elbow method computes the inertia of the cluster model, which is the sum of the

distances of samples to their closest cluster center (SSE), in order to establish the optimal number of clusters.

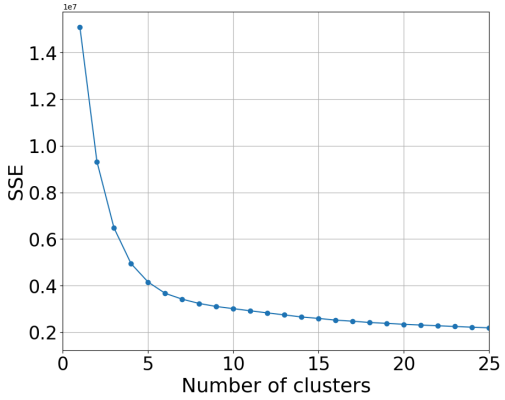

Figure 3. The number of clusters defined by the Elbow method for the rotating machine dataset.

By analyzing Figure 3, a number of 5 clusters was selected to monitor the rotating machine. This point on the curve represents the maximum distance between the line connecting the first plotted point (number of clusters = 1) and the last plotted point (number of clusters = 25). Afterwards, the initial clustering model is defined using approximately 40 seconds of measured data. Subsequently, each new state is computed by incorporating an increment of approximately 2 seconds of newly measured data. Figures 4, 5, 6, 7, and 8 illustrate the evolution of the Jaccard Overlap, stability, relative population, centroid walking distance, and centroid distance from the initial state metrics for the cluster that changed the most during the monitoring of this rotating machine. The black line represents the rotating machine during normal functioning, while the blue dotted, green dash-dotted, and red dashed lines represent different levels of unbalance faults. Additionally, the vertical magenta dashed line indicates the moment when the fault conditions began to occur.

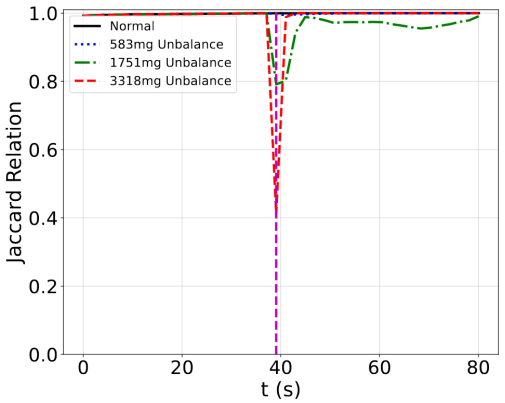

Figure 4. The Jaccard Overlap metric computed over time for the rotating machine.

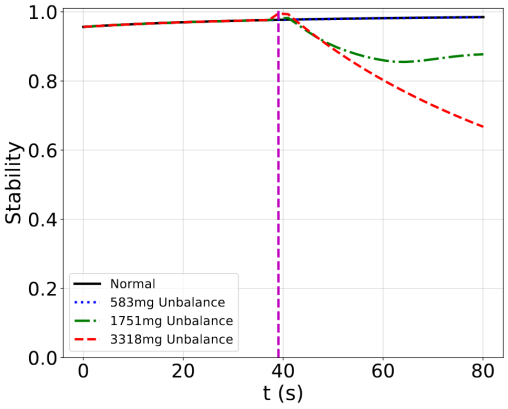

Figure 5. The stability metric computed over time for the rotating machine.

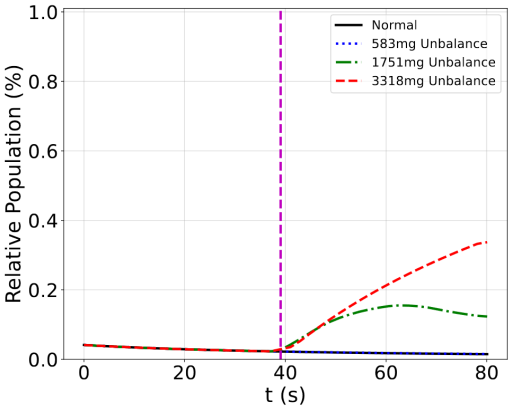

Figure 6. The relative population metric computed over time for the rotating machine.

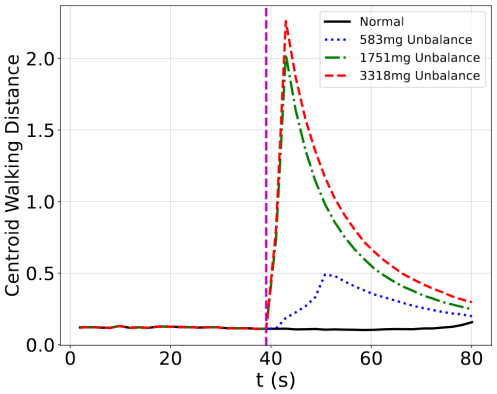

Figure 7. The centroid walking distance metric computed over time for the rotating machine.

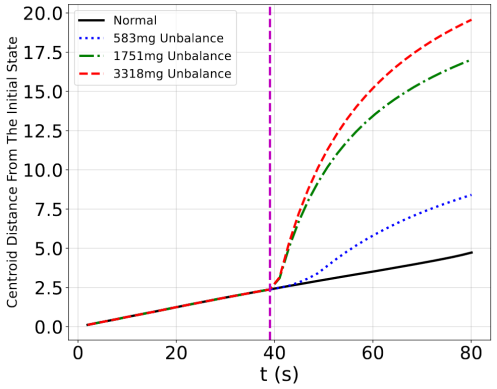

Figure 8. The centroid distance from the initial state metric computed over time for the rotating machine.

By analyzing Figures 4-8, it is evident that when the faults begin to occur, the change in this cluster's behavior is very noticeable and is immediately indicated by the Jaccard Overlap metric, which suggests that changes in the machine's functioning occurred during the transitioning state. While this change in functioning is constant and continues to occur, metrics such as relative population, stability, and centroid distance from the initial state keep increasing until they reach a new level. Additionally, it can be observed that when the fault begins, the cluster centroid starts to walk (see Figure 7) until it reaches a new quasi-steady state on new coordinates. Another interesting observation is that the magnitude of clustering change is directly proportional to the severity of the fault, making it easier to detect severe faults.

ENGINE DATASET - In Figure 9, the Elbow method is depicted to determine the number of clusters to be defined by the K-means algorithm for the engine dataset.

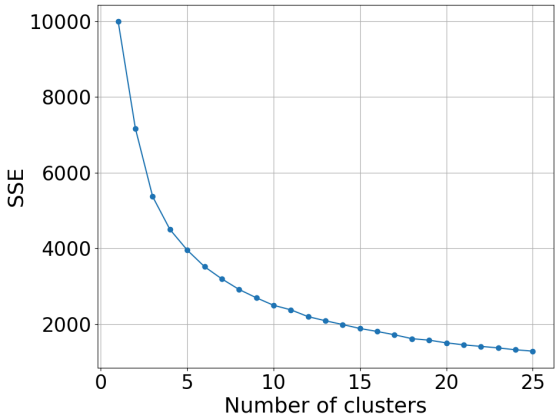

Figure 9. The number of clusters defined by the Elbow method for the engine dataset.

By analyzing Figure 9, a number of 6 clusters was selected for monitoring the vehicle's ICE. This point on the curve represents the maximum distance between the line connecting the first plotted point (number of clusters = 1) and the last plotted point (number of clusters = 25). Afterwards, the initial clustering model is defined using approximately 7 minutes of measured data. Subsequently, each new state is computed by incorporating an increment of approximately 10 seconds of newly measured data. Figures 10, 11, 12, 13, and 14 illustrate the evolution of the Jaccard Overlap, stability, relative population, centroid walking distance, and centroid distance from the initial state metrics for the cluster that changed the most during the monitoring of this vehicle's ICE. The black line represents the ICE during normal functioning, while the blue dotted line represents the ICE working with a simulated fault condition. Additionally, the vertical red dashed line indicates the moment when the simulated fault condition began to occur.

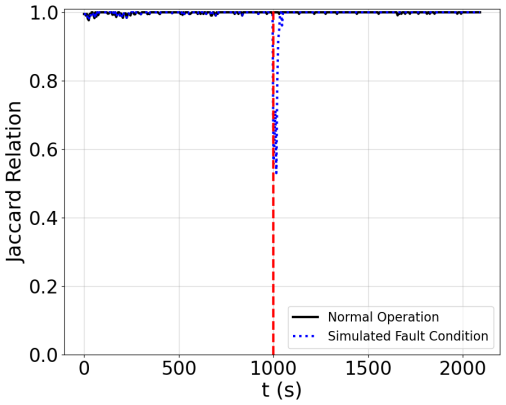

Figure 10. The Jaccard Overlap metric computed over time for the rotating machine.

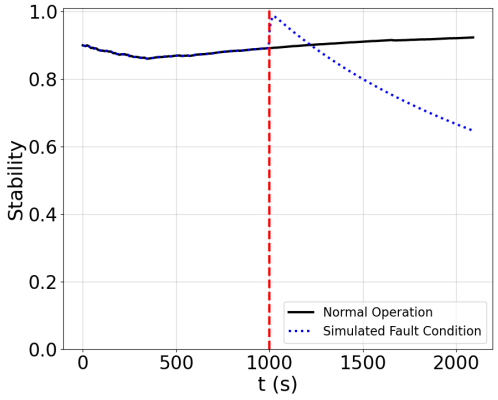

Figure 11. The stability metric computed over time for the rotating machine.

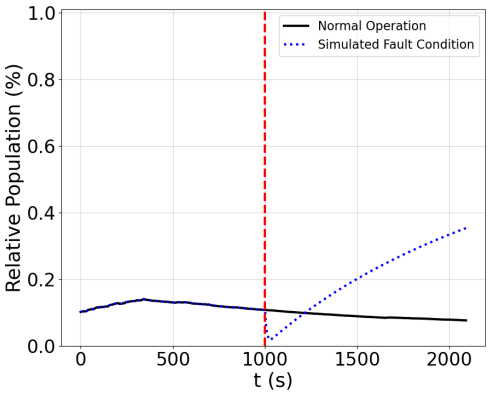

Figure 12. The relative population metric computed over time for the rotating machine.

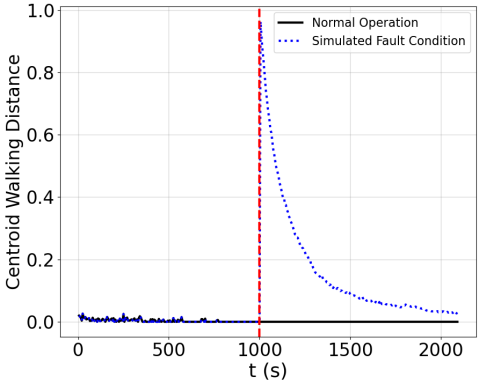

Figure 13. The centroid walking distance metric computed over time for the rotating machine.

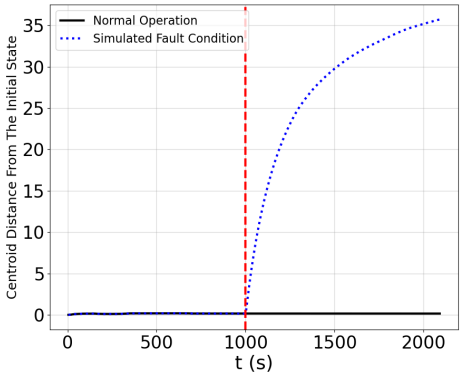

Figure 14. The centroid distance from the initial state metric computed over time for the rotating machine.

Upon analysis of Figures 10-14, mirroring the findings from the previous dataset, it becomes apparent that when the simulated fault initiates, there is a pronounced alteration in the behavior of this cluster. This change is readily discernible and promptly indicated by the Jaccard Overlap metric, implying shifts in engine functionality during the transitional state. While this functional shift persists consistently, metrics such as relative population, stability, and centroid distance from the initial state continue to fluctuate until reaching a new equilibrium. Furthermore, it is observed that with the onset of a fault, the cluster centroid begins to transition (refer to Figure 13), eventually stabilizing at a new quasi-steady state with revised coordinates. These results underscore the efficacy of the proposed approach in swiftly detecting genuine faults, enabling automated alerts to signal system health deterioration through threshold-based alarms, potentially adjusted by the weight delineated in Eq. 5. The pipeline depicted in Figure 2 can thus be leveraged to integrate an Internet of Things (IoT)-based solution aimed at autonomously monitoring the health of the vehicle's ICE.

CONCLUSIONS

The present work proposes an IoT-based solution incorporating an unsupervised ML method to automatically monitor the health of a vehicle's ICE. This approach was validated using a public dataset of an unbalanced rotating machine and tested on an idling medium-sized vehicle's ICE by simulating faults with the air conditioning system. Various monitoring metrics are presented, such as Jaccard Overlap, stability, relative population, centroid walking distance, and centroid distance from the initial state, revealing significant behavior changes for all considered fault scenarios.

The results are promising, demonstrating that the proposed evolutionary metrics approach has the potential to detect actual faults and could be utilized for automated health monitoring of the considered systems. Importantly, this solution could be extended to other dynamical systems, as it operates independently of physics-based information, relying solely on data. However, a limitation of the current approach is its inability to classify faults. Nonetheless, it could serve as a trigger for other models aimed at classifying such faults. Moreover, further experiments should be conducted to explore higher volumes of data and real engine faults.

In this context, the solution pipeline—encompassing data collection from a datalogger device, communication with a middleware IoT platform, and data processing and monitoring via unsupervised ML clustering—represents a powerful tool for aiding the health monitoring of vehicle ICEs.

REFERENCES

[1] Henriquez P, Alonso J B, Ferrer M A, Travieso C M. Review of Automatic Fault Diagnosis Systems Using Audio and Vibration Signals. IEEE Transactions on Systems, Man and Cybernetics: Systems, 2013.

[2] Zhang Q, Zuo Z, Liu J. Failure analysis of a diesel engine cylinder head based on finite element method, Engineering Failure Analysis, vol 34, 2013.

[3] Liu X F, Wang Y, Liu W H. Finite element analysis of thermo-mechanical conditions inside the piston of a diesel engine, Applied Thermal Engineering, vol 119, 2017.

[4] Ghorpade U S, Chavan D S, Patil V, Gaikwad M. Finite element analysis and natural frequency optimization of engine bracket, International Journal of Mechanical and Industrial Engineering, vol 3, 2013.

[5] Seralathan S, Mitnala S V, Reddy RV S K, Venkat I G, Reddy D R T, Hariram V, Premkumar T M. Stress analysis of the connecting rod of compression ignition engine, Materials Today: Proceedings, vol 33, 2020.

[6] Avwunuketa A, Enyia J, Oloruntoba S. Numerical and Thermal Finite Element Analysis (FEA) of Idealized Gas Turbine Engine Blade, International Journal of Aerospace and Mechanical Engineering, vol 7, 2020.

[7] Sun L, Shang Z, Bhowmick S, Nagarajaiah S. Review of Bridge Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection, Journal of Structural Engineering, vol 146, 2020.

[8] Zhao R, Yan R, Chen Z, Wang P, Gao R X. Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing, 2019.

[9] Dong S, He K, Tang B. The fault diagnosis method of rolling bearing under variable working conditions based on deep transfer learning. Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol 42, 2020.

[10] Zhang Q, Lin J, Song H, Sheng G. Fault Identification Based on PD Ultrasonic Signal Using RNN, DNN and CNN. Condition Monitoring and Diagnosis (CMD), 2018.

[11] Liang P, Deng C, Wu J, Yang Z, Zhu J. Intelligent Fault Diagnosis of Rolling Element Bearing Based on Convolutional Neural Network and Frequency Spectrograms. IEEE International Conference on Prognostics and Health Management, 2019.

[12] Ramteke S M, Chelladurai H, Amarnath M. Diagnosis and Classification of Diesel Engine Components Faults Using Time-Frequency and Machine Learning Approach. Journal of Vibration Engineering & Technologies, 2021.

[13] Carrera F, Falchi F, Girardi M, Messina N, Padovani C, Pellegrini D. Deep learning for structural health monitoring: An application to heritage structures. Cornell University Electrical Engineering and Systems Science, 2022.

[14] Junqueira B F, Leiderman R, Castello D A. Damage recovery in composite laminates through deep learning from acoustic scattering of guided waves. Ultrasonics, vol 139, 2024.

[15] Xiong M, Wang H, Fu Q. Digital twin–driven aero-engine intelligent predictive maintenance. The International Journal of Advanced Manufacturing Technology, vol 114, 2021

[16] Jan S U, Lee Y D, Koo I S. A distributed sensor-fault detection and diagnosis framework using machine learning. Information Sciences, vol 547, 2021.

[17] Chukwudi I J, Zaman N, Rahim M A, Rahman M A, Alenazi M J F, Pillai P. An Ensemble Deep Learning Model for Vehicular Engine Health Prediction. IEEE Access, vol 12, 2024.

[18] Liao J, Hu J, Yan F, Chen P, Zhu L, Zhou Q, Xu H, Li J. A comparative investigation of advanced machine learning methods for predicting transient emission characteristic of diesel engine. Fuel, vol 350, 2023.

[19] Khurana S, Saxeng S, Jain S, Dixit A. Predictive modeling of engine emissions using machine learning: A review. Materials Today: Proceedings, vol 38, 2021.

[20] Xu X, Zhao Z, Xu X, Yang J, Chang L, Yan X, Wang G. Machine learning-based wear fault diagnosis for marine diesel engine by fusing multiple data-driven models. Knowledge-Based Systems, vol 190, 2020.

[21] Tang M, Chen H, Guan C. Research on diesel engine fault diagnosis method based on machine learning. 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC), 2022.

[22] Ramteke S M, Chelladurai H, Amarnath M. Diagnosis and Classification of Diesel Engine Components Faults Using Time–Frequency and Machine Learning Approach. Journal of Vibration Engineering & Technologies, vol 10, 2021.

[23] Junqueira B F, Violato R P V, Simões F O, Tuleski B. Aplicação de Machine Learning para Diagnóstico de Falha em Motores a Diesel utilizando Sinais de Áudio. Anais do XXIX Simpósio Internacional de Engenharia Automotiva, 2022.

[24] Aliramezani M, Koch C R, Shahbakhti M. Modeling, diagnostics, optimization, and control of internal combustion engines via modern machine learning techniques: A review and future directions. Progress in Energy and Combustion Science, vol 88, 2022.

[25] Landauer M, Wurzenberger M, Skopik F, Settanni G, Filzmoser P. Time series analysis: Unsupervised anomaly detection beyond outlier detection. In C. Su & H. Kikuch (Eds.), Information Security Practice and Experience, 2018.

[26] Niwattanakul S, Singthongchai J, Naenudorn E, Wanapu S. Using of Jaccard Coefficient for Keywords

Similarity. Proceedings of the International MultiConference of Engineers and Computer Sc, vol 1, 2013.

[27] Greene D, Doyle D, Cunningham P. Tracking the evolution of communities in dynamic social networks. International Conference on Advances in Social Network Analysis and Mining, 2010.

[28] Toyoda M, Kitsuregawa M. Extracting Evolution of Web Communities from a Series of Web Archives. Association for Computing Machinery, 2003.

[29] Zhou A, Cao F, Qian W, Jin C. Tracking clusters in evolving data streams over sliding windows. Knowledge and Information Systems, vol 15, 2008.

[30] Jung W, Kim S H, Yun S H, Bae J, Park Y H. Vibration, acoustic, temperature, and motor current dataset of rotating machine under varying operating conditions for fault diagnosis. Data in Brief, vol 48, 2023.

[31] Lee J, Kim J, Bae C. Effect of the air-conditioning system on the fuel economy in a gasoline engine vehicle. Proceedings of the Institution of Mechanical Engineers, 2013.