An Ensemble Learning-Based Alert Trigger System for Predictive Maintenance of Assets with In-Situ Sensors

DAN AO[1], GUGA GUGARATSHAN[1], DAKOTA BARTHLOW[1], SARAH MIELE[1], JERI BOUNDY[1] and JAMES WARREN[2]

ABSTRACT

Failure of assets, such as machines, engines, or equipment, can cause significant loss to an enterprise. Real-time monitoring of the asset operation and early detection of failure could guide mechanics or engineers to check the assets in time and reduce the chance of a breakdown. With the availability of Internet of Things (IoT) technologies and the development of machine learning, predictive maintenance has become an effective way to monitor asset performance and detect anomalies using sensor data and historical information. However, many use cases need more information to evaluate the results. Lack of validation can undermine confidence in predictive results. To solve this problem, a novel alert trigger schema that integrates sensor fusion, feature extraction, machine learning, ensemble strategy, and an alert format is proposed. The method offers a comprehensive and reliable approach to detecting anomalies based on unlabeled data, utilizing sensor and decision-level fusion techniques. Meanwhile, eight anomaly detection techniques were investigated, including K-Means Clustering, Gaussian Mixture Model, Autoencoder, and Isolation Forest. Various algorithms generate results with differing degrees of confidence. These results are consolidated into a single indicator representing the alert level. This amalgamation of data ensures the provision of robust and reliable predictions. Instead of simply combing the alert information, confidence in different algorithms is reflected in adding different weights in the ensemble process. In addition, while other existing frameworks focus on evaluating the algorithm's accuracy, more effort was put into demonstrating a level-based alert system, showing diagnosis changing with time. In that way, mechanics or engineers can get information to check the asset's status. The proposed framework was validated using a synthetic dataset based on recordings from a rotating fault simulator that generates multi-modal data, including accelerometer, acoustic, and tachometer data, representing the run state of the rotating components. The alert system showed different levels of warning for predictive maintenance. The framework is designed with high flexibility and scalability. Therefore, this framework can be generalized to other in situ-sensor data from various assets.

---

[1]Hottinger Bruel & Kjaer Solutions LLC, 100 Research Blvd., Suite 216, Starkville, MS 39759
[2]Mississippi State University, 75 B. S. Hood Rd, Mississippi State, MS 39762

INTRODUCTION

Failure of critical assets, such as machines, engines, or equipment, can cause significant loss to an enterprise. It is crucial to detect defects early and implement proper maintenance to minimize the aftermath of system malfunctions. Traditionally, routine maintenance is made to help keep equipment operating smoothly. This is referred to as preventive maintenance. Preventive maintenance is typically scheduled using empirical rules and statistical data, which could include data on the estimated lifespan of each component or may be based on the occurrence of previous component failures. However, in preventative maintenance, the actual internal operating conditions and external environmental factors are often overlooked [1], leading to a possibility of over or underestimating the necessary period for upkeep. Predictive maintenance (PdM) provides a promising solution to all these problems.

PdM also known as "online monitoring," "risk-based maintenance," or "condition-based maintenance," is a maintenance approach that uses asset conditions as a basis for scheduling maintenance tasks [2]. Through real-time asset operation monitoring, we preemptively take measures to prevent failures and only schedule maintenance tasks when necessary [3, 4]. Contemporary PdM techniques use data-driven approaches and incorporate various advanced methods derived from data management and machine learning. Machine learning approaches are effectively applied in areas with much sensor data. Machine Learning-based PdM can be broadly categorized into two main classes: supervised and unsupervised. In supervised PdM, the dataset used for modeling contains failure information, while in unsupervised PdM, there is only logistic or process information to be utilized [5]. In this work, we intend to develop an alert trigger system for predictive maintenance based on unlabeled data. Here unsupervised machine learning method is used to distinguish normal and anomalous behavior of the target feature or component. This is commonly known as anomaly detection.

Anomaly detection refers to “the problem of finding patterns in data that do not conform to expected behavior” [6, 7]. Detecting anomalies is a critical tool in various applications, such as intrusion detection for cyber-security, fraud detection for credit cards, monitoring of medical conditions, and fault detection for aviation safety studies. The advent of Internet of Things (IoT) technologies has made collecting vast amounts of sensor data possible. Consequently, more sophisticated machine learning algorithms have been developed to leverage historical data to capture the system's behavior. In machine learning, anomaly detection is often categorized as unsupervised learning. Standard anomaly detection algorithms include K-Means Clustering, Isolation Forest, Autoencoder, etc. However, each algorithm has its limitations. A lack of benchmark data could make it hard to select the optimal algorithm and undermine confidence in predictive accuracy. To solve this problem, a novel alert trigger schema that integrates sensor fusion, feature extraction, machine learning, ensemble strategy, and an alert format is proposed. This method offers a comprehensive and reliable approach to detecting anomalies based on unlabeled data, utilizing sensor and decision-level fusion techniques. Including multiple anomaly detection techniques and using weights based on confidence level ensure accurate and robust predictions. The emphasis on a level-based alert system enables mechanics or engineers to monitor the asset's status over time, making it a valuable tool for PdM.

The proposed framework was validated using a synthetic time series dataset generated based on recordings from a rotating fault simulator to represent run state of the rotating components. Multi-modal data including accelerometer, acoustic, and tachometer data are collected. By integrating an alert system, the framework exhibited varying levels of warnings for PdM. Furthermore, its high flexibility and scalability enable it to be easily adapted to other in situ-sensor data from diverse assets.

The remainder of this work is organized as follows: Section 2 provides a brief review of the anomaly detection algorithms relevant to this study. Section 3 presents a detailed description of the proposed schema. In Section 4, the effectiveness of the proposed approach is demonstrated on the fault simulator dataset. Finally, Section 5 offers concluding remarks and discusses future research plans pertaining to this topic.

RELATED WORKS

Our work involves implementing eight commonly used unsupervised anomaly detection algorithms to identify anomalies. This section focuses on briefly reviewing and evaluating these anomaly detection techniques. Popular anomaly detection techniques include cluster-based algorithms such as K-Means Clustering, Gaussian Mixture Model, and DBSCAN. These methods group data points based on their similarities, either by distance (K-Means Clustering) or density (Gaussian Mixture Model and DBSCAN). Outliers are determined by how far they extend from a cluster group. Another approach to detect anomalies is by using transformation techniques such as Principal Component Analysis (PCA) and Autoencoder. In these methods, data points with reconstruction loss above a set threshold are identified and labeled as anomalous. It is worth noting that there are some anomaly detection algorithms that derived from supervised learning techniques but are used for unsupervised learning. One example is the Isolation Forest anomaly detection machine learning algorithm, which is developed from Random Forest and uses a tree-based approach to isolate anomalies. Unlike most model-based anomaly detection approaches that profile normal instances and identify instances that do not conform to the normal profile, Isolation Forest better captures the key essence of anomalies: the concept of "few and different" [8]. Another example is the one-class Support Vector Machine (SVM) algorithm, which involves a single class of data points, and the task is to predict a hypersphere that separates the cluster of data points from the anomalies. And lastly, the Mahalanobis Distance is a highly effective multivariate distance metric that quantifies the distance between a point and a distribution. It is commonly used to identify data points that are significantly different from the normal distribution profile.

METHODOLOGY

In the previous section, we discussed some popular anomaly detection algorithms, but each algorithm has its own limitations. For instance, K-Means Clustering may not work well when clusters have irregular shapes [9], Isolation Forest may only detect local anomaly point [10], and Autoencoders may lose important information in the input data

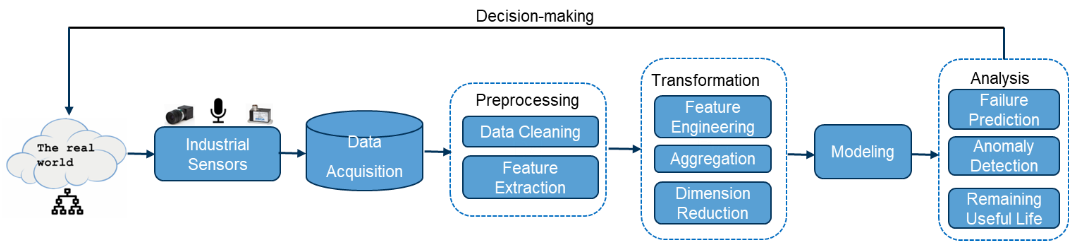

Figure 1. Visualization of the data analysis pipeline

due to their sensitivity to input errors. The performance of an algorithm depends on the underlying data structure. To achieve reliable and robust, a novel alert trigger schema is proposed that combines sensor fusion, feature extraction, machine learning, ensemble strategy, and an alert format.

This project aims to detect anomalies in unlabeled multi-variate time series data by utilizing sensor fusion and decision level fusion techniques. In the proposed alert trigger system, different types and formats of data, are preprocessed and integrated into a single Data Frame. Next, a couple of steps such as feature extraction, and dimension reduction may be required to transform the data. Once the data is in a unified view, it can be fed into a model for analysis, such as failure prediction, anomaly detection, and Remaining Useful Life (RUL) analysis. Figure 1 illustrates the top-level block diagram of the data analysis pipeline. It is important to note that each type of analysis may require a different model and specific preprocessing steps tailored to the problem at hand.

The crucial part of the alert trigger system is the decision-level fusion. The results of eight popular anomaly detection algorithms with varying degrees of confidence are merged into a single indicator and then normalized to an anomaly score. To enhance the ensemble process, different weights are assigned to each algorithm based on its confidence level. Moreover, the anomaly scores are divided into four categories representing different trigger levels. The emphasis on a level-based alert system enables mechanics or engineers to monitor the asset's status over time. The general process for applying the system to a dataset is shown in Figure 2.

Figure 2 illustrates that each anomaly detector labels the data instance as either 0 (no anomaly detected) or 1 (anomaly detected). The normalized anomaly score at a particular instance is then obtained through a weighted average of the labels from

Schema of the Trigger System

Anomaly Detector	**Outlier Assessment**										
K-means Clustering	0	0	…	0	1	…	1	1	…	1	1
Gaussian Mixture Model	0	0	…	0	0	…	0	1	…	1	1
DB Scan	0	0	…	0	0	…	1	1	…	1	1
Autoencoder	0	0	…	0	0	…	1	1	…	1	1
PCA	0	0	…	0	0	…	0	0	…	0	1
Isolation Forest	0	0	…	0	0	…	0	0	…	1	1
OneClassSVM	0	0	….	0	1	….	0	1	…	1	1
Mahalanobis Distance	0	0	…	1	0	…	1	1	…	1	1
Normalized Score	**0**	**0**	**…**	**0.125**	**0.25**	**…**	**0.5**	**0.75**	**…**	**0.875**	**1**
Trigger	**T0**	**T0**	**…**	**T1**	**T1**	**…**	**T2**	**T2**	**…**	**T3**	**T3**

Ensemble:
Weighted Indicator
Averaging

Trigger Level
T0: Healthy
T1: Check
T2: Service
T3: Fail

Figure 2. Schema of ensemble learning-based alert trigger system

multiple anomaly detectors, as described below:

$$Score = \left(\sum_{i=1}^{n} w_i f_i\right) \Big/ \left(\sum_{i=1}^{n} w_i\right) \tag{1}$$

where for a total of n anomaly detectors, w_i is the user defined weight assigned for the i'th detector, and f_i is the label obtained from the i'th detector. Note that the range of the normalized score is between 0 and 1. When the score is 0, we consider the target component or system is healthy (T0). Otherwise, the level of alert is triggered following with the user selected rule below:

$$Trigger = \begin{cases} T1, & 0 < Score \leq 1/3 \\ T2, & 1/3 < Score \leq 2/3 \\ T3, & 2/3 < Score \leq 1 \end{cases} \tag{2}$$

CASE STUDY

The effectiveness of the proposed framework was validated using a synthetic time series dataset generated based on recordings from a rotating test bench fault simulator. This section describes the experimental setup, outlines the process for generating training and testing data, and presents the results of the evaluation.

Experimental Setting

To gain a comprehensive understanding of different vibration signatures, controlled experiments were conducted using a Machinery Fault Simulator (MFS)[11] that emulates real-world industrial machinery. According to our research interests, various test kits are available for data collection. In this project, we used the following configuration in MFS: a Centrally Balanced Rotor with faults induced at the 3/4" Shaft Bearing (including No Fault, Ball Fault, Inner Race Fault, Outer Race Fault, and Combination Bearing Fault), a rotational speed of 1800 rpm, and a sample rate of 10k. We collected signals from three types of sensors: microphones (8), accelerometers (3 for "XYZ" axis), and a tachometer (1). Each run lasted 60 seconds, with 6 runs for the Healthy condition and 3 runs for each of fault condition. This configuration enabled us to collect a comprehensive dataset for analysis and testing. Figure 3 shows the MFS experimental setup and configuration.

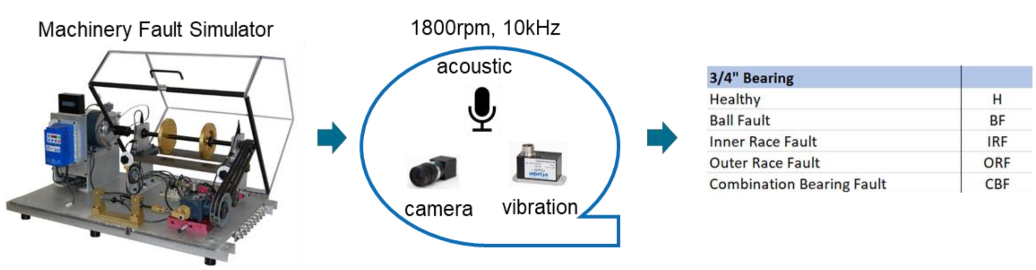

Figure 3. Experimental setup and data collection

Dataset

Training and testing data were synthesized based on the recordings. Considering that the healthy signals are stationary and pure, we contaminated a portion of normal signals with abnormal instances for model training. For the test dataset, by interleaving periods of time series from healthy and unhealthy simulated data, the team created a 3 second miniature sample dataset representative of the run state of an asset. Figure 4 displays a sample for an acoustic data channel. In this plot, initially the response is stationary (similar to the healthy dataset), which is attributed to normal asset performance. Next, the asset signal begins to display a deviation from normal operation. As the time passes, issues in the asset starts to increase and at the end of the select sample we can visually detect a failure.

Results

The effectiveness of the proposed framework was validated using the dataset discussed earlier. The performance of each algorithm relies on the underlying data structure, which, in our case, is closely linked to the failure mode. As a result, our confidence (reflected on weights) in each algorithm may differ based on the failure mode concealed within the testing data. When Ball Fault was employed as the failure type, we observed that the Autoencoder, Gaussian Mixture Model, and Isolation Forest algorithms were better able to identify outliers, resulting in a higher weight assignment. In this case, these three algorithms were assigned the weight “2”, while other algorithms were given weight “1”. Note the choice of weights are user dependent and can be changed upon based on application of the technique.

In our case, we obtain predictions from eight anomaly detectors at each time step, which are integrated into a normalized anomaly score and then categorized to three trigger levels using the rule described in Equation 2. The Excel file generated by the ensemble trigger system for inspection is structured as shown in Figure 2 and can be visualized as demonstrated in Figure 5. In Figure 5, the top signal represents the acoustic channel, while the eight dotted lines depict the anomalies detected by the eight anomaly detectors. The anomaly detectors, arranged from top to bottom, include K-means Clustering (KM), Gaussian Mixture Model (GMM), DB Scan (DB), Autoencoder (ATE), PCA, Isolation Forest (IF), Support Vector Machine (SVM), and Mahalanobis

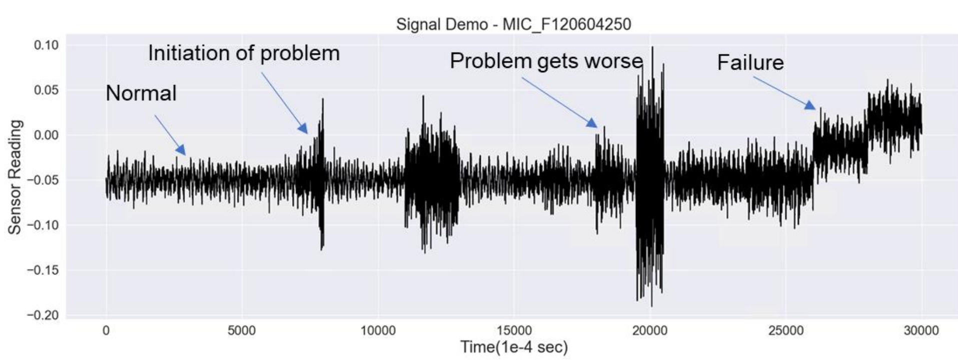

Figure 4. Synthetically generated signal representing asset normal life, failure initiation, and failure

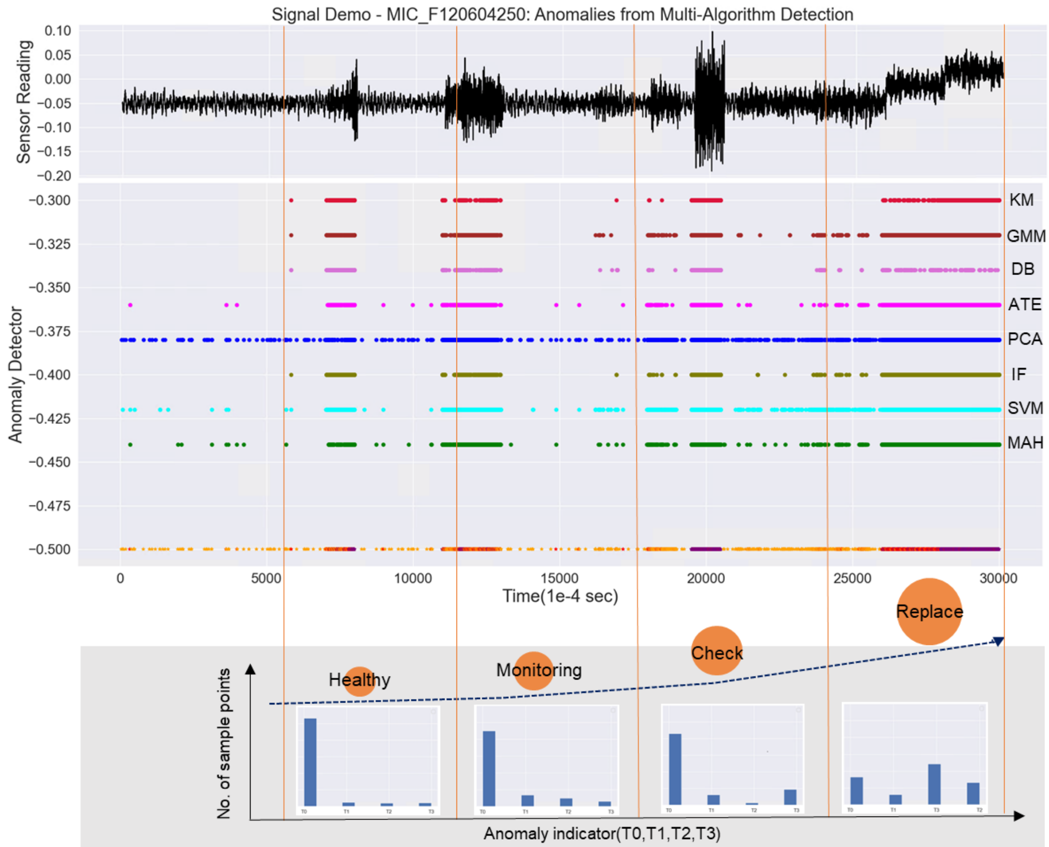

Figure 5. A Visualization of Alert Trigger System

Distance (MAH). Although the predictions generally align, there are slight variations among them. The star line below indicates the different levels of alert from the ensemble model, with "orange," "red," and "purple" representing increasing levels of alertness. Towards the bottom, a bar chart is generated for each 0.6-second window, starting from 0.6 seconds. This bar plot illustrates the anomaly indicator (trigger levels) and their concentrations. Within each bar chart, the bars are labeled T0, T1, T2, and T3, from left to right. In the histogram, the majority of the samples are located at T0. However, subsequent plots reveal an increasing number of samples moving towards T1, T2, and T3. Notably, in the final plot, high concentrations of anomalies are observed in T2 and T3. This series of bar plots demonstrates how the asset progresses from a normal state to failure. Based on the distribution of trigger levels, inspectors can make informed decisions regarding the condition of the asset.

CONCLUSIONS

This study presents an ensemble-learning based alert trigger system that integrates sensor fusion, feature extraction, machine learning, ensemble strategy, and an alert format is presented. The framework is designed with high flexibility and scalability. We applied this procedure to a synthetic dataset generated from MSF recordings to evaluate its effectiveness. Using this alert trigger system, we can produce an Excel file containing diagnoses from eight anomaly detectors and the ensemble model to inform predictive maintenance. Although we have only used acoustic, accelerometer, and tachometer data due to the lack of image data, we plan to incorporate image data in the future to demonstrate its ability to process different types of data.

The data collected from MSF are drawn at a high frequency of 10kHz, making it impractical to conduct health monitoring directly on the raw data. Moving forward, we intend to simulate an extended dataset and leverage the recorded datasets as labeled samples of failures. This will enable us to evaluate the effectiveness of our system in detecting failures.

ACKNOWLEDGEMENTS

This material is based upon work supported by the ERDC under Contract No. W912HZ23C0013. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ERDC. OPSEC approval reference FY23-0001 W912HZ23C0013.

REFERENCES

1. Namuduri, S., et al., *Deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors.* Journal of The Electrochemical Society, 2020. **167**(3): p. 037552.
2. Paolanti, M., et al. *Machine learning approach for predictive maintenance in industry 4.0*. in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. 2018. IEEE.
3. Lughofer, E. and M. Sayed-Mouchaweh, *Predictive maintenance in dynamic systems: advanced methods, decision support tools and real-world applications*. 2019: Springer.
4. Susto, G.A., et al., *Machine learning for predictive maintenance: A multiple classifier approach.* IEEE transactions on industrial informatics, 2014. **11**(3): p. 812-820.
5. Wang, K., *Intelligent predictive maintenance (IPdM) system–Industry 4.0 scenario.* WIT Transactions on Engineering Sciences, 2016. **113**: p. 259-268.
6. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey*. ACM computing surveys (CSUR), 2009. **41**(3): p. 1-58.
7. Injadat, M., et al. *Bayesian optimization with machine learning algorithms towards anomaly detection*. in *2018 IEEE global communications conference (GLOBECOM)*. 2018. IEEE.
8. Xu, H., et al., *Deep isolation forest for anomaly detection.* IEEE Transactions on Knowledge and Data Engineering, 2023.
9. Singh, K., D. Malik, and N. Sharma, *Evolving limitations in K-means algorithm in data mining and their removal.* International Journal of Computational Engineering & Management, 2011. **12**(1): p. 105-109.
10. Gao, R., et al. *Research and improvement of isolation forest in detection of local anomaly points*. in *Journal of Physics: Conference Series*. 2019. IOP Publishing.
11. Machine Fault Simulator: https://spectraquest.com/machinery-fault-simulator/details/mfs/