Matthew C. Kersting[a], Ajay M. Patrikar[a], Eric Schneider[a], Terry Kusunoki-Martin[a], Alexander Drumm[b], Casey O'Neill[b]
[a]Noblis; [b]Naval Sea Systems Command (NAVSEA)

Condition-Based Maintenance Using Unsupervised Time-Series Anomaly Detection

ABSTRACT

United States maritime services are looking to better integrate artificial intelligence (AI) into their maintenance procedures to improve readiness. Modern naval ships employ many onboard sensors to collect information about the ship's systems, performance, and navigation parameters. The current method for analyzing this data and detecting possible issues is mostly manual. The abundance of sensor data enables one to apply machine learning (ML) techniques to continuously monitor and analyze the ship's operations. ML systems can be applied to automatically detect or predict and report potential failures. Using ML to automate the analysis of shipboard sensors will allow the Navy to move from time-based maintenance to condition-based maintenance. In this paper, we describe experiments in which the sensor data is represented as multivariate time-series. In practice, the anomalies do not occur frequently, and the data associated with anomalies is scarce. Therefore, we apply unsupervised time-series anomaly detection (UTSAD) techniques to find anomalies. Initially, the experiments are carried out on a simulated time-series with artificially inserted anomalies, so that the ground truth is known, and quantitative results can be obtained. We experiment with point anomalies as well as subsequence anomalies such as shift, trend, and variance anomalies. For the detection of anomalies, we present results using a density-based spatial clustering of applications with noise (DBSCAN) method, a tree-based isolation forest (IF) method, and a reconstruction-based autoencoder (AE) method. Finally, we present results on the actual ship data and discuss the implementation of an onboard real-time hierarchical analytics system.

Keywords: Unsupervised machine learning, multivariate time-series, anomaly detection, outlier detection, preventive maintenance, predictive maintenance

INTRODUCTION

The health and maintenance of marine gas turbine engines are pivotal in the operational readiness of the U.S. Navy surface fleet. Marine gas turbine engines are used to generate electrical power for the vessel, as well as propulsion power. Currently, the maintenance of these gas turbines is largely manual, with maintenance actions occurring on a routine basis or on an as-needed basis. Simultaneously, modern naval ships employ many onboard sensors to collect information about the ship's performance and navigation parameters from the vessels' machinery and ship control systems. The current method for analyzing this data and detecting possible issues is, again, largely manual. The abundance of sensor data enables one to apply AI and ML techniques to continuously monitor and analyze the ship's operations. ML systems can be applied to automatically detect or predict and report potential failures. Automating this process can free operators from repetitive tasks and potentially reduce the possibility of error as well as potentially mitigate a larger catastrophic failure.

The Navy has a requirement to improve the readiness and availability of ships and systems in a budget-constrained environment. Using ML to automate the analysis of shipboard sensors will allow the Navy to move from time-based maintenance to condition-based maintenance and ultimately a predictive maintenance model, meaning that maintenance can be performed when the condition of the ship dictates a maintenance schedule rather than a fixed, and sometimes reactive, schedule. Figure 1 shows the steps in the desired transition.

The large amount of sensor data collected from naval ships can be leveraged in moving toward condition-based maintenance. The relevant data is collected from a variety of sensors onboard the ship, and these sensors produce values at somewhat unpredictable intervals of time. Based on the signal definition, sensor sampling can range anywhere from 1Hz to what is driven by the machine state. The available sensors may be sufficient for the control of operations but not necessarily for maintenance prediction. These sensor values are not directly labeled with a failure event. The lack of labeled data makes us treat this as an unsupervised time-series anomaly detection problem. With this approach, we can fully leverage the abundance of sensor data without relying on the laborious task of labeling each data point as normal or assigning it a failure mode.

Time-series data can be analyzed in either a univariate or a multivariate manner. Given the higher dimensionality of the data in multivariate approaches, they tend to be more challenging; however, it is possible for an algorithm to analyze the correlation between sensor values. Anomalies can be categorized as point or subsequence anomalies; a point anomaly is a data point that deviates significantly from normal behavior, and a subsequence anomaly is a group of data points that collectively deviate from normal behavior. In this study, both types of anomalies are analyzed.

The paper is organized as follows. Section 2 briefly describes related work on time-series anomaly detection, including its use in analyzing operations of marine engines. Section 3 describes our steps towards finding solutions to the problems of predictive diagnostics and prognostics of marine engines. In Section 4, experimental results are presented on artificial as well as real ship data. Section 5 presents our approach toward the onboard implementation of our selected algorithms. Finally, Section 6 gives a summary and conclusions.

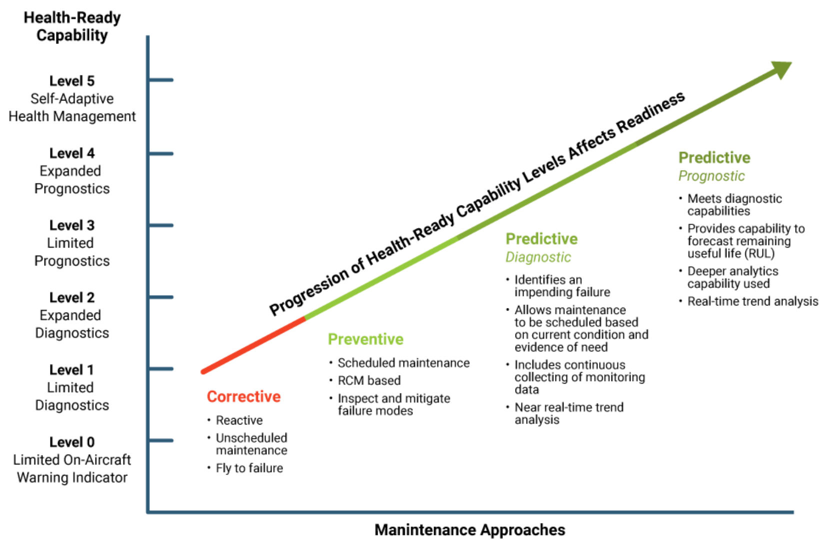

Figure 1. The U.S. Navy is moving towards predictive maintenance procedures which are prognostic in nature. (Source: Brief to Deputy Assistant Secretary of the Navy on Condition-Based Maintenance.)

RELATED WORK

Detecting anomalous patterns in time-series has been a very active area of research due to its applications in many areas such as finance, fraud detection, healthcare monitoring, diagnostics, manufacturing processes, etc. Since the time-series data can be large and complex, researchers have developed various specialized algorithms for the automatic detection of anomalous behavior. The surveys of these techniques can be found in [1]-[3]. This problem has been approached from many different angles by the research communities belonging to areas such as signal analysis, stochastic learning, statistics, data mining, classical ML, and deep learning [2]. Broadly speaking, three distinct learning schemes can be applied to multivariate time-series anomaly detection tasks: supervised, semi-supervised, and unsupervised. Supervised learning assumes that a fully labeled dataset is available for learning concrete boundaries between anomalous and regular data. Semi-supervised learning assumes that only some input data are labeled. Unsupervised learning identifies hidden data patterns from unlabeled datasets. The difficulty with unsupervised techniques tends to be in the evaluation of those models [3].

Data analysis and ML techniques have been applied for the health and performance monitoring of gas turbine engines [4]-[6]. Additionally, ML has been applied as a method for the predictive analysis of marine engines. This predictive analysis has included predicting propulsion power [7], energy efficiency [8], and fuel consumption [9]-[10]. When applying ML to predictive maintenance, it has been noted that there are strong correlations among sensors from the same piece of equipment and that key parameters generally have a strong influence on the overall machinery [11]. Several unsupervised anomaly detection methods have been applied to monitor the marine engine. These include a cluster-based approach [12], a method based on auto-associative kernel regression [13], a method based on statistical process control [14], and an ensemble-based method that can deal with high-dimensional and large-scale sensor data [15]. Recently, Kim et al. [16] applied the isolation forest algorithm to detect anomalies in naval sensor data and then applied explainable AI techniques to understand which sensors are deviating significantly from their normal operation.

METHODOLOGY

As we stated above, modern ships have a control system network that may have control sufficiency but is not robust enough for maintenance prediction. This is why the current method for analyzing this data and detecting possible issues is largely manual or at best "deterministic" by hard-set alarm bands or a time threshold. The abundance of sensor data enables one to apply onboard analytics to continuously monitor and analyze the ship's operations. As part of our methodology, we developed a system hierarchal framework by sensor clusters to determine "boundaries" where anomalies may be indicated. The analytics modules aided by the ML algorithms can be applied to automatically detect or predict and report potential failures.

Hierarchical framework

The marine gas turbine engine is a complicated mechanical system consisting of several component systems as shown in Figure 2. We believe that processing and analyzing data from all the component systems simultaneously is a difficult task because of the complexity of the problem. Therefore, we propose a framework in which there is an analytics module (AM) for each component system, and all the anomalies are aggregated at the main level with a centralized analytics module (CAM). In this paper, we focus on building an AM for the fuel system while keeping such a framework in mind.

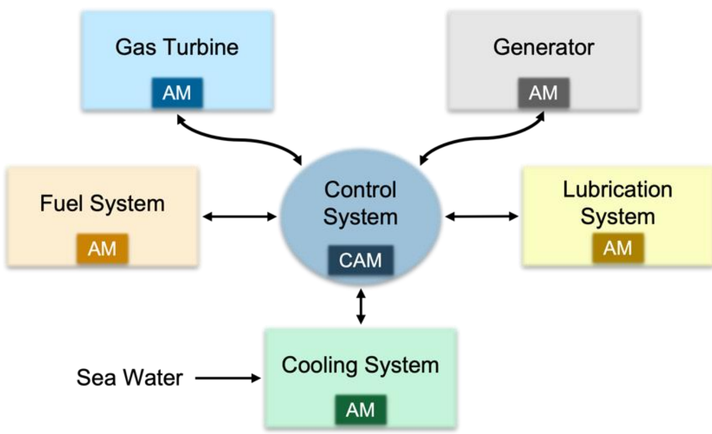

Figure 2. A simplified system diagram of a marine gas turbine generator. It is desirable to set up a hierarchical framework consisting of a centralized analytics module (CAM) that oversees the analytics modules (AMs) of the component systems.

Artificial versus real data

Most machine learning systems are data-driven. The nature of the data plays an important role in the selection of algorithms and the quality of data has a major impact on their performance. For the purpose of this paper, we obtained sensor and maintenance data from various Guided-Missile Destroyer (DDG) vessels. While there have been attempts to collect and annotate data from such vessels in the last few years, the process is not yet mature. We made the following observations about the available data: (a) while it is possible to collect sensor data at regular sampling intervals, the available data had many gaps, and (b) the maintenance records contained notes from the technicians written in plain English which sometimes have inconstancies in the maintenance action write up. It was not always clear if the maintenance action was triggered by a faulty operation, or if it was carried out as a part of routine maintenance and inspection. Also, only domain experts have sufficient knowledge to determine if a given maintenance event could have been forecasted with the help of sensors. For these reasons, the evaluation of the algorithms on the real data can only be subjective in nature.

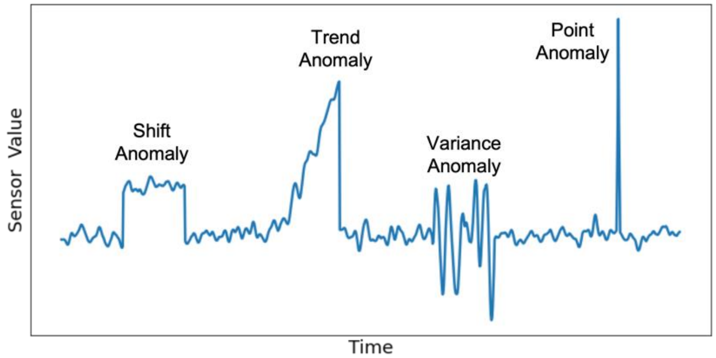

Figure 3. Simulated time-series with artificial anomalies.

To obtain quantitative results to evaluate the efficacy of algorithms, we decided to first test them on a simulated time-series data in which artificial anomalies were injected at known times. For this purpose, we generated multivariate time-series data using autoregressive moving average processes and smoothing functions and injected four types of anomalies: (1) shift anomalies, (2) trend anomalies, (3) variance anomalies, and (4) point anomalies, as shown in Figure 3. While examining the available real data, it was not possible for us to visually identify the anomalous behavior of various sensors. Therefore, we do not know if the artificial anomalies are indicative of the real anomalies. Despite that, we designed our algorithms and set thresholds in such a way that we can detect most artificial anomalies and expect that those algorithms would also be effective in detecting real anomalies, whatever kind they might be.

Algorithm selection

A majority of the UTSAD methods fall under the following four categories [2]: (1) clustering-based methods, (2) forecasting-based methods, (3) tree-based methods, and (4) reconstruction-based methods. Clustering-based methods use distance metrics to compare time-series points or subsequences with each other. Time-series values that are farther away from the normal clusters are considered outliers or anomalies. We apply a popular clustering method called density-based spatial clustering of applications with noise (DBSCAN) [17]-[18] to this problem. Tree-based methods successively split data using binary trees and identify anomalies as data that get rapidly isolated in the trees. We apply a popular tree-based method called isolation forest (IF) [16, 19]. Forecasting-based methods use a learned model to forecast time-series values based on a context window. Anomalous behavior is detected if the predicted values show a significant deviation from the observed values. Forecasting-based methods require time-series values at regular sampling intervals. Since this is not true for the available data, we rule out the use of forecasting methods. Reconstruction-based methods build a model of normal behavior by encoding subsequences of a normal training time-series in a low-dimensional latent space. A decoder then reconstructs time-series values from the latent space. A large error between reconstructed values and observed values implies anomalous behavior. We experiment with a reconstruction method based on an autoencoder (AE) [22]-[23], a block diagram of which is shown in Figure 4. Brief descriptions of the selected algorithms are in Table 1.

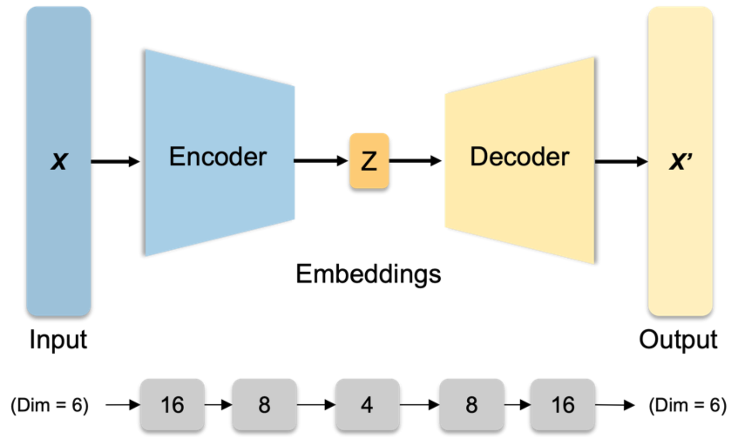

Figure 4. Autoencoder model. In our experimentation, we use dense layers with 16 and 8 units for the encoder and 8 and 16 units for the decoder, respectively. The dimensionality of our input and output is 6, and there are 4 latent variables.

Table 1. Anomaly detection algorithms selected for experimentation.

Algorithm	**Description**
DBSCAN	DBSCAN is a density-based clustering algorithm proposed in 1996 [18]. It is a non-parametric algorithm that groups data points that are closely packed together. The data points that lie alone in low-density regions are considered outliers. There are two key parameters of DBSCAN: *epsilon* and *minPts*. Two data points are neighbors if the distance between them is less than or equal to *epsilon*. *minPts* is the minimum number of data samples required to define a cluster. Based on these two parameters, data points are classified as core points, border points, or outliers. A data point is a core point if there are at least *minPts* number of samples in its surrounding area with radius *epsilon*. A point is a border point if it is reachable from a core point and there are less than *minPts* number of points within its surrounding area. A point is an outlier if it is not a core point and not reachable from any core points.
Isolation forest	Isolation forest is a tree-based anomaly detection method first proposed in 2008 [19]. Since then, there have been several improvements suggested for the algorithm [20, 21]. During the training phase, IF starts with a root node consisting of all the data points. While building a tree, every internal node is split into two sub-nodes until there is complete data isolation or maximal tree depth is reached. Data is considered isolated when it is alone in its node. During the scoring phase, a score for each data point is calculated which reflects the similarity degree between the test data point and other items in the tree. The number of nodes crossed by the test data point from the root node to reach its external node is called the path length. Data points with shorter path lengths are likely to be anomalies.
Autoencoder	Autoencoder [22, 23] is a deep learning-based model consisting of two major components, an encoder, and a decoder, as shown in Figure 4. The encoder component projects the input data onto a latent embedding space and the decoder component takes data from the embedding space and reconstructs value in the original space. Autoencoders can be used to detect anomalies because, during training, the model learns the data and begins to reconstruct it accurately, but the reconstruction error happens to be larger for unusual data points. Given this assumption, we

	flag data points as anomalous if they have an unusually high reconstruction error. Also, by monitoring reconstruction errors for individual sensors, we can often determine which of the sensors are showing anomalous behavior. This feature provides a degree of explainability which many other algorithms are unable to provide.

EXPERIMENTAL RESULTS

We first performed experiments on the simulated data with artificial anomalies. Once the algorithms were confirmed to show acceptable results on the simulated data, we performed experiments on the actual data acquired from the ships.

Results on artificial data

To be able to evaluate the algorithms quantitatively, we created a tool to generate multivariate time-series data with artificially injected anomalies. To emulate the sensors from the fuel system, we simulated a multivariate time-series with six variables. Each variable is generated using an independent autoregressive moving average process with different parameters, followed by smoothing functions with different window sizes. We first generated data without anomalies and set it aside for training. We then generated test data and inserted point, shift, trend, and variance anomalies of various amplitudes. Our test data contained 34 anomalies, 24 of which are single-channel anomalies and 10 are two-channel anomalies in which there were position overlaps between anomalies. An example of our test data is shown in Figure 5.

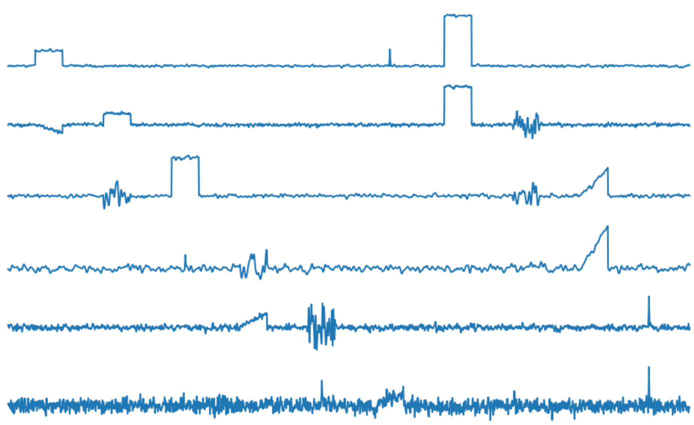

Figure 5. An example of a simulated multivariate time-series with artificially injected anomalies.

All algorithms are sensitive to parameter settings. In our experiments, we attempted to find parameters through trial and error to maximize the performance of each algorithm. The receiver operating characteristics (ROC) of the three algorithms are shown in Figure 6. To obtain a ROC chart for the isolation forest algorithm, we varied the contamination parameter from 0.0005 to 0.06 in steps of 0.005. For the DBSCAN algorithm, we set *minPts* at 12, but varied *epsilon* from 0.1 to 0.7 in steps of 0.1. Our autoencoder architecture is shown in Figure 4. To obtain a ROC curve for the autoencoder, we changed the threshold on the mean-squared error from 0.1 to 20.0 in the steps of 0.5.

The false-alarm rate in Figure 6 is defined as the percentage of normal data points which were classified as outliers. In practice, a single data point would not be declared anomalous. Instead, we would be looking for several adjacent outlier data points occurring within a time window. Thus, the false-alarm rates for all algorithms are likely to be substantially lower than what is shown in Figure 6. Nonetheless, this figure is indicative of the effectiveness of the three algorithms.

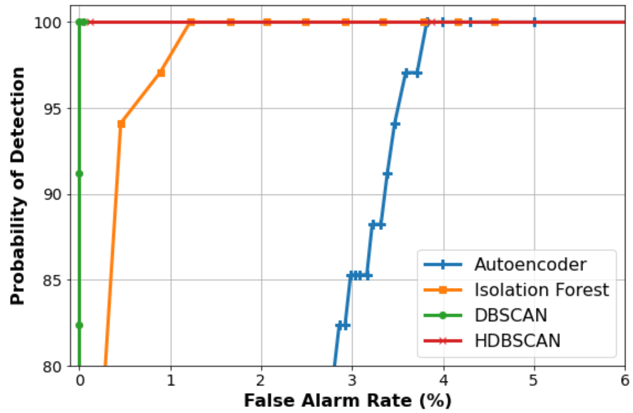

Figure 6. Performance of DBSCAN, isolation forest, and autoencoder algorithms on artificial data.

DBSCAN is the top-performing algorithm. However, DBSCAN is computationally very expensive because during inference, every new data point must be compared with tens of thousands of DBSCAN clusters that are formed to determine if it is an outlier. Therefore, it may not be feasible to run the DBSCAN algorithm in real time on the ship's computer. The computations associated with the other two algorithms are reasonable, so it should be possible to implement them in real-time without significant computational resources. The performance of the IF algorithm appears to be much better than the AE algorithm. However, the AE algorithm has one significant advantage over the other two methods because it can indicate which of the input sensor(s) are showing anomalous behavior. AE algorithm allows monitoring of reconstruction error for individual sensors, which means it can identify one or more sensors that are displaying anomalous behavior. Ship's maintenance personnel may find this feature valuable for diagnostic purposes.

Combining algorithms for improved performance

The selected algorithms have very different decision-making mechanisms. Therefore, there is value in combining the outputs of these algorithms to improve overall accuracy and explainability. For example, the IF can be the default algorithm which is run in real time and can act as a pre-screener. Whenever the IF algorithm produces an alarm, we can run the DBSCAN algorithm only on those data points to determine if the DBSCAN algorithm concurs with IF. If it does, then we can run the autoencoder to determine which sensors have the highest reconstruction errors thus identifying individual sensor(s) with anomalous behavior.

Results on real data

Several assumptions were made while generating the artificial data in terms of the types, durations, and magnitudes of the anomalies. The data acquired from the ship is quite complex and we do not have the ability to examine the data visually and determine if those assumptions are valid. Therefore, we cannot say with certainty that the algorithms successful on the artificial data would also be successful on the actual data. For this reason, we implemented all three algorithms to generate anomalies for the real data. The anomalies/alarms generated by these algorithms would be evaluated subjectively by the ships' maintenance experts to determine which algorithms are effective.

The actual ship data provided to us was from 24 ships, each one of them having three engines for a total of 72 engines. For 13 engines either the maintenance or sensor data was not available, so that left us with data from 59 engines which we could use for algorithm development and analysis. The data was collected over a period of six years and was from the historical database, meaning it was not 1Hz data, but trend data stored over time hourly. An example of sensor data and maintenance events for one engine over a period of approximately six years is shown in Figure 7. It shows that the historical sensor data is not available continuously and may not always correspond with the maintenance events. Maintenance records contain information about when the maintenance began and ended along with the description of the parts replaced and/or the procedure. Without domain knowledge, it is not possible for us to determine if the maintenance event could have been detected with the help of sensors.

For training the anomaly detectors, we combined data from all the engines, and trained DBSCAN, IF, and AE algorithms on that data without the use of any labels. We then set thresholds for each in such a way that about one percent of the data points are classified as anomalous by the algorithms. We apply anomaly detection algorithms to the data; generate alarms and produce plots for each engine as shown in Figure 7. The maintenance subject matter experts (SME) would determine the effectiveness of each algorithm by examining the generated alarms and their confidence value and deciding if they are true detections or false alarms. Based on a sampling of the detected anomalies and maintenance records, a positive correlation was found between the detection of an anomaly and the indication of a need for maintenance.

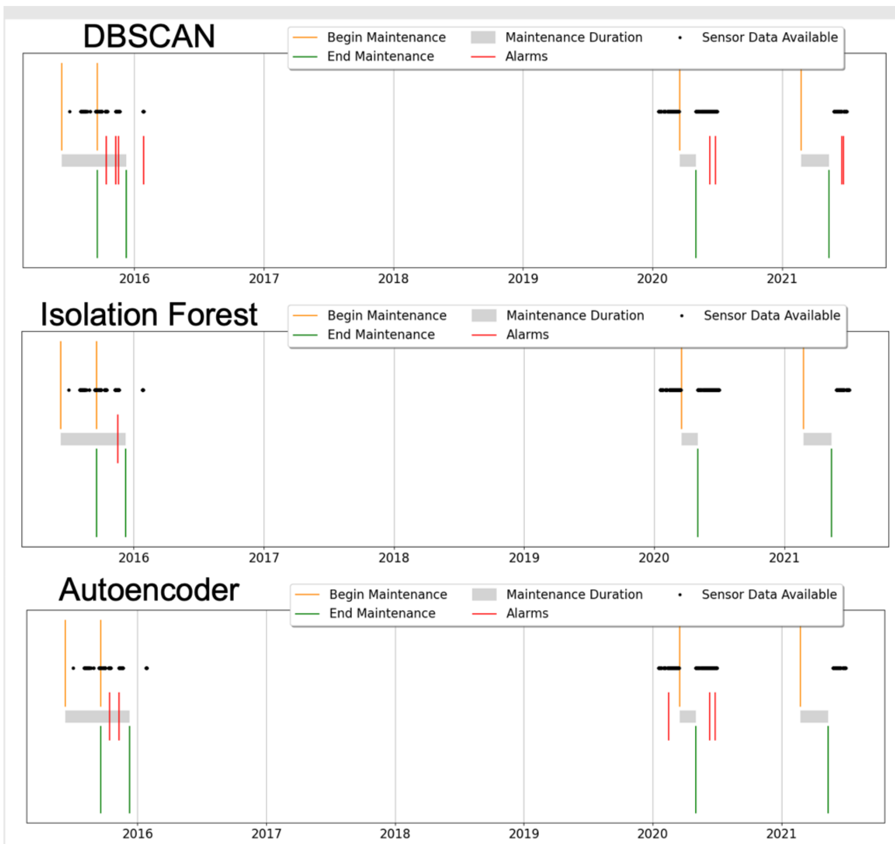

Figure 7. Alarms generated by DBSCAN, isolation forest, and autoencoder for a specific engine.

IMPLEMENTATION

After the algorithms are trained and evaluated, they are then deployed to begin ingesting and analyzing data in real time. Data ingestion in the production environment is more challenging than during training and evaluation as is often the case with ML projects. To effectively deploy the trained models, some data wrangling first needs to be performed. In the production environment, a subset of the sensors available in the training data produces signals at irregular intervals. The sensors are not producing output simultaneously. So, to run inference, a batch of recent sensor values from all applicable sensors is collected. Then the timings of the sensor values are matched while unmatched values are dropped. The batch of collected values is normalized and anomalies are detected. Afterward, a waiting period of the batch size multiplied by the typical interval between sensor values is performed before fetching new values. The batch size is then a tradeoff, longer batch sizes lead to fewer inference runs; however, this also potentially leaves a longer period between the production of an anomalous value and its analysis.

Detected anomalies are then reported to the crew in the form of generic maintenance actions. In the case of reconstruction-based algorithms, such as the AE, we can report more details about the anomaly to help inform the crew. Since reconstruction-based methods can report a reconstruction error per sensor value, we are able to report to the crew the specific sensors that are behaving most anomalously. This is a form of AI explainability that does not significantly affect inference speed.

The deployed application is written in Python using common ML libraries like scikit-learn and TensorFlow. After performing extensive unit testing, integration testing, and static code analysis, the application is containerized for deployment and orchestration by Kubernetes. The core logic of anomaly detection and reporting is abstracted away from any data sources by using "connectors" which are responsible for integrating with a particular problem space. Future development will include online learning, so as new values are collected, the algorithm's distribution of normal data can be adapted.

CONCLUSIONS

In this paper, we have presented a design of a hierarchical analytics framework that can be used to continuously monitor a ship's operation by examining the outputs of onboard sensors. The framework leveraged three algorithms that use very different mechanisms for anomaly detection. At the time of this paper, we have conducted several SME reviews on sampled anomalies detected and maintenance actions, with positive confidence in our approach. The next step in this research will be to deploy a live instance on a marine vessel operating on 1 Hz data. We have shown that there is a potential to combine these algorithms to achieve better accuracy and explainability. Automating this process can free operators from repetitive tasks and potentially reduce the possibility of error. Based on some of our initial reviews, it is possible to flag the anomaly with adequate time to plan repairs, ask for assistance, or change operation prior to a failure event, thereby increasing the machine uptime and readiness. These are all steps towards predictive maintenance of naval ships which would facilitate the readiness and availability of ships in a budget-constrained environment. The final system would require onboard implementation of such a framework and thorough testing over an extended period.

REFERENCES

[1] Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021). A review on outlier/anomaly detection in time series data. ACM Computing Surveys (CSUR), 54(3), 1-33.

[2] Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment, 15(9), 1779-1797.

[3] Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2022). Deep Learning for Time Series Anomaly Detection: A Survey. arXiv preprint arXiv:2211.05244.

[4] Tahan, M., Tsoutsanis, E., Muhammad, M., & Karim, Z. A. (2017). Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review. Applied energy, 198, 122-144.

[5] Maraini, D., Simpson, M., Brown, R., & Poporad, M. (2018, September). Development of a Data-driven Model for Marine Gas Turbine (MGT) Engine Health Monitoring. In Annual Conference of the Prognostics and Health Management Society. URL: https://pdfs.semanticscholar.org/4e85/6fc68c11d3260763669864177be79c8513b1.pdf

[6] Poporad, M., Viercinski, J., Caguiat, D., Brown, R., Maraini, D., Simpson. M. (2019). Marine gas turbine fault detection using a data-driven healthy engine model. Proceedings of ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition GT2019 June 17-21, 2019, Phoenix, Arizona, USA.

[7] Kim, D., Lee, S., & Lee, J. (2020). Data-driven prediction of vessel propulsion power using support vector regression with onboard measurement and ocean data. Sensors, 20(6), 1588.

[8] Beşikçi, E. B., Arslan, O., Turan, O., & Ölçer, A. I. (2016). An artificial neural network based decision support system for energy efficient ship operations. Computers & Operations Research, 66, 393-401.

[9] Gkerekos, C., Lazakis, I., & Theotokatos, G. (2019). Machine learning models for predicting ship main engine Fuel Oil Consumption: A comparative study. Ocean Engineering, 188, 106282.

[10] Mak, L., Sullivan, M., Kuczora, A., & Millan, J. (2014, September). Ship performance monitoring and analysis to improve fuel efficiency. In 2014 Oceans-St. John's (pp. 1-10). IEEE.

[11] Jimenez, V. J., Bouhmala, N., & Gausdal, A. H. (2020). Developing a predictive maintenance model for vessel machinery. Journal of Ocean Engineering and Science, 5(4), 358-386.

[12] Vanem, E., & Brandsæter, A. (2021). Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine. Journal of Marine Engineering & Technology, 20(4), 217-234.

[13] Brandsæter, A., Vanem, E., & Glad, I. K. (2019). Efficient on-line anomaly detection for ship systems in operation. Expert Systems with Applications, 121, 418-437.
[14] Bae, Y. M., Kim, M. J., Kim, K. J., Jun, C. H., Byeon, S. S., & Park, K. M. (2018). A case study on the establishment of upper control limit to detect vessel's main engine failures using multivariate control chart. Journal of the Society of Naval Architects of Korea, 55(6), 505-513.
[15] Kim, D., Lee, S., & Lee, J. (2020). An ensemble-based approach to anomaly detection in marine engine sensor streams for efficient condition monitoring and analysis. Sensors, 20(24), 7285.
[16] Kim, D., Antariksa, G., Handayani, M. P., Lee, S., & Lee, J. (2021). Explainable anomaly detection framework for maritime main engine sensor data. Sensors, 21(15), 5200. URL: https://www.mdpi.com/1424-8220/21/15/5200
[17] Kirichenko, L., Radivilova, T., & Tkachenko, A. (2019). Comparative Analysis of Noisy Time Series Clustering (Doctoral dissertation).
[18] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).
[19] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In Eighth IEEE international conference on data mining (pp. 413-422). IEEE.
[20] Hariri, S., Kind, M. C., & Brunner, R. J. (2019). Extended isolation forest. IEEE Transactions on Knowledge and Data Engineering, 33(4), 1479-1489.
[21] Marteau, P. F., Soheily-Khah, S., & Béchet, N. (2017). Hybrid isolation forest-application to intrusion detection. arXiv preprint arXiv:1705.03800.
[22] Sakurada, M., & Yairi, T. (2014, December). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proc. of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis (pp. 4-11).
[23] Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. arXiv preprint arXiv:2003.05991

AUTHOR BIOS

Matthew Kersting is a Machine Learning Engineer and Software Developer at Noblis. He is experienced at developing machine learning solutions from prototypes to robust and scalable applications. Matthew has been responsible for driving research projects in time-series analysis, computer-vision, and data management. He has a master's degree in biomedical engineering from the University of South Florida.

Ajay M. Patrikar, Ph.D., is a Director/Senior Scientist at Noblis. He has more than 25 years of experience developing machine learning solutions for audio, images, video, radar, and RF data. His research interests include responsible AI, computer vision, deep neural networks, statistical pattern recognition, image and video processing, and speech and speaker recognition. He received his Ph.D. degree in Electrical Engineering from Southern Methodist University.

Eric Schneider, P.E., is a Program Manager and Lead Solutions Architect at Noblis. He has over 28 years of engineering experience in maintenance and modernization, including condition monitoring program support for NSWC PD, NAVSEA and fleet forces command. He has over 20 years of contractual management experience both commercially and with the Navy, including management of Logistics support contracts with NSWC PD supporting System Health Assessment and leading several internal research projects on predictive sparing, ILS certification automation. He received his bachelor's degree in mechanical engineering from Old Dominion University.

Terry Kusunoki-Martin is a Machine Learning Engineer and Software Developer at Noblis. He has 7 years of experience doing development work for various DHS projects and has been the project lead for internal machine learning research efforts. He received his bachelor's degree in computer science from Swarthmore College.

Alexander Drumm is a data scientist at Naval Sea Systems Command.

Casey O' Neill is a Machine Learning Engineer and Software Developer at Naval Sea Systems Command. He has a master's degree in applied mathematics from Columbia University.