Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-16 (2021) 105–111

Anomaly Detection with Unknown Anomalies: Application to Maritime Machinery

Katarzyna Michałowska * **Signe Riemer-Sørensen** ** **Camilla Sterud** *** **Ole Magnus Hjellset** ****

* *University of Oslo & SINTEF, Oslo, Norway (e-mail:katarzyna.michalowska@sintef.no)*
** *SINTEF, Oslo, Norway (e-mail:signe.riemer-sorensen@sintef.no)*
*** *SINTEF, Oslo, Norway (e-mail:camilla.sterud@sintef.no)*
**** *Brunvoll, Molde, Norway (e-mail:ole.magnus.hjellset@brunvoll.no)*

Abstract: We present a framework for deriving anomaly detection algorithms on timeseries data when the time and expression of anomalous behaviour is unknown. The framework is suited for problems in which individual machine learning paradigms cannot be directly implemented: supervised learning is not applicable due to the lack of labelled data, unsupervised learning is not effective since the normal operations are insufficiently defined and take complex and diverse forms, and deep learning risks confusing problematic behaviours for expected ones due to the possible repetitiveness of similar anomalies. The proposed approach is comprised of two phases: unsupervised discovery of anomalies, and semi-supervised construction and tuning of the anomaly detection algorithm. By leveraging data exploration methods and expert knowledge, the resulting algorithms are interpretable and detect a wide range of anomalous behaviours. The approach is applied to the early detection of wear and tear of maritime propulsion and manoeuvring machinery. We show that the final algorithm is able to detect different types of anomalies, including an actual internal leakage in a thruster which is otherwise overlooked by the present rule-based alarm system.

Keywords: anomaly detection, predictive maintenance, machine learning, condition-based monitoring, fault detection and diagnosis, grey-box modelling.

1. INTRODUCTION

Automated condition monitoring in complex physical systems requires flexible algorithms that are able to detect a wide range of irregularities indicative of malfunction. Through the ability to find complex patterns in the data, machine learning methods can detect less pronounced and less predictable deviations from regular operation than e.g., rule-based monitoring systems, usually suited to identify specific problems.

Machine learning algorithms are typically divided into two categories: supervised and unsupervised. In the supervised approach, such as in Kim et al. (2020), the algorithm learns representations of anomalous and normal behaviours based on labels corresponding to each sample e.g., the normal and abnormal class, or a value indicative of a deviation. In reality, labelled data with representative anomalies are usually not available in quantities sufficient to learn such representations, especially since anomalies can take many forms, which makes supervised methods inapplicable. In unsupervised approaches, anomalies are detected as outliers in data that are otherwise considered to represent normal conditions (Vanem and Brandsæter, 2019). These methods are often preferred even when labelled data are available, as anomalous states are usually underrepresented (Tan et al., 2020). However, the use of unsupervised learning requires a well-defined normal behaviour and identification of informative features that are able to discern symptoms of malfunction from e.g., different operational modes or unusual environmental conditions. If these conditions are not met, unsupervised algorithms exhibit low performance.

To circumvent the need for manual feature engineering, many works propose the use of deep autoencoders, which belong to the semi-supervised category (Kong et al., 2021; Zhang et al., 2019). These may not be appropriate for condition monitoring, as anomalies which are symptomatic of machinery malfunction are expected to be similar and repetitive, therefore they may be missed by the autoencoder that conceptualizes recurring patterns as normal behaviour. In addition, autoenconders require great amounts of data representing all the normal modes of operation and do not provide interpretable metrics (Pang et al., 2021).

In this work, we propose a two-phase framework for constructing anomaly detection algorithms without labelled data and sufficiently defined normal behaviour, as presented in Fig. 1. The first phase is anomaly discovery which incorporates data exploration and expert knowledge. This is reminiscent of active learning, i.e., interactive

10.1016/j.ifacol.2021.10.080

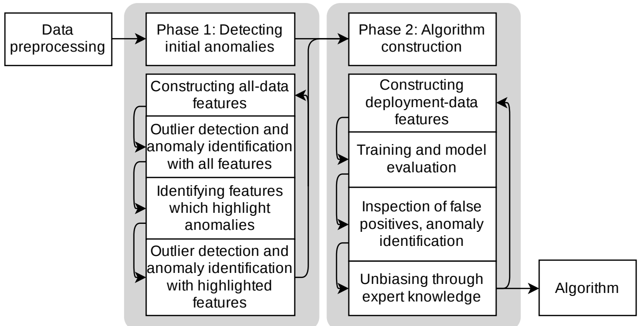

Fig. 1. Framework pipeline.

labelling of new anomalous instances using expert feedback (Das et al., 2016; Beaugnon et al., 2017; Görnitz et al., 2012), though the methods are suited for discovering diverse anomalies to model informative features, rather than for labelling data to train a supervised algorithm. In the second phase, the final algorithm is constructed, tuned with the sparse anomalous labels and prevented from overfitting through the inclusion of expert knowledge. Depending on the application, the final algorithm can use human-interpretable features that allow for analysis of the detected anomalies.

The paper is organized as follows: Section 2 introduces the general framework, Section 3 presents the use case of a propulsion and manoeuvring system, to which the framework is applied in Section 4. Section 5 presents the performance of the derived algorithm on the basis of its ability to detect a variety of anomalies, while Section 6 derives the main conclusions of the work.

2. FRAMEWORK FOR DISCOVERY AND DETECTION OF ANOMALIES

In this section, we describe each phase of the framework leading to the construction of an anomaly detection algorithm, as summarized in Fig. 1. Phase 1 is an iterative process of feature engineering and data exploration for detecting outliers and identifying anomalies in the training data. In phase 2, the most informative features are used to construct the anomaly detection algorithm and expert knowledge is included to avoid bias towards specific anomalies. The transition criterion from one phase to another is purposefully undetermined: Phase 1 continues until a diverse array of anomalies is found, and phase 2 until the detection algorithm achieves satisfactory performance.

2.1 Pre-processing

The suggested framework utilizes data divided into equal-length time series, where each time series is a sequence of multiple measurements treated as an independent observation. This approach reduces the data that needs to be analysed by summarizing the intervals with consideration of time dependencies, which requires much less computation than analysing each point individually. Although dividing the data into smaller periods disrupts its continuity and may cause some time-dependent interactions to be lost, an appropriate interval length and overlap can alleviate this problem. The intervals should be longer than the time scale of variation in the data, but shorter than the time we expect the observed system to stay in an anomalous state. Physical anomalies are expected to persist, therefore they should be detected through multiple consecutive time intervals.

The pre-processing procedure depends on system details. In this work, we consider the following steps:

- Resampling: Sensor data is often gathered under an event-triggered sampling regime, such as Lebesgue sampling. Resampling is applied to emulate an even sample rate and align data from different sensors;
- Missing value imputation: If a value is not available at some time within the resampling interval, the last available measurement is typically used;
- Correcting values to ensure physical consistency;
- Outlier removal: Time series with infeasible values are discarded.

2.2 Phase 1: Detecting anomalies in the training data

The goal of the first phase is to use unsupervised data exploration techniques, outlier detection and manual verification of the resulting outliers to identify anomalous states. It is unclear how the normal and anomalous operations are expressed in the data, therefore the first step is to understand relationships that highlight the differences between them. The goal of the initial anomaly detection is to aid feature engineering, rather than identify all anomalies.

Oftentimes anomalies are more pronounced in data that may not be available in deployment, due to sensor limitations. In the first phase, we identify anomalous periods using all available data, and aim to understand patterns that occur in the final-model data during these periods.

Below we highlight some considerations with regard to feature engineering:

Variable interactions: Though some anomalies may be identifiable by one variable alone, multi-variable interactions are typically more informative. Visual analysis can identify relationships in the data and be a basis for deriving new features and detecting anomalies. The analysis can consist of raw data plots, pairwise variable comparisons, and density plots. Data exploration should be aided with domain knowledge whenever possible.

Leveraging variable dependencies: State estimation can reveal changes in underlying assumptions through an increased deviation of measurement and estimate. The estimate can be based on other measurements available in the data.

Automatized feature extraction: The data can be analysed either as individual timestamps or as summarized characteristics of time series. In the latter, the feature extraction can be a combination of expert knowledge, and automatized feature extraction tools, such as *tsfresh* in Python (Christ et al., 2018).

Dimensionality reduction: In cases with many variables, or when a large set of descriptive statistics is produced, the data can be transformed using dimensionality reduction techniques, such as principal component analysis

(PCA) or t-distributed stochastic neighbour embedding (t-SNE) (Pedregosa et al., 2012). Such techniques aim to retain the most information while discarding noise. However, in some cases the anomalous behaviour may be lost in the transformation, especially if the dimensionality is significantly reduced. Similarly, removal of correlated variables may result in removing information that highlights anomalous behaviour.

Initially, detected anomalies can help identify features that capture their characteristics, and conversely, these features can be used to detect more anomalies. Identification of informative features can be conducted by comparing feature distributions for samples of anomalous and non-anomalous observations. One option is a two-sample Kolmogorov-Smirnov test (Dodge, 2008) with the null hypothesis that the two samples (anomalous and non-anomalous) come from the same distribution. The features with the lowest p-values are then used to detect anomalies.

Relevant features are leveraged, either individually or with data reduction techniques, to detect further outliers through thresholds on feature values, outlier-aware clustering methods, or outlier detection algorithms such as isolation forest and one-class support vector machines.

2.3 Phase 2: Iterative tuning of the detection algorithm

The second phase can be conceptualized as semi-supervised learning. The problem is posed as supervised anomaly detection, as the initial anomalies are already found, but an unsupervised algorithm is used due to the proportion of anomalies in the data and that some anomalies remain undiscovered. In the tuning process, some proportion of false positives is desired, as long as it indicates discoveries of new anomalies. Simultaneously, the aim is to maximise the number of true positives, while maintaining a low number of false negatives. In contrast to the first phase, only sensor measurements that are available in deployment are used.

Initially, the algorithm is trained on samples without known anomalies. Here we propose one-class SVM (Section 4.4), but clustering methods or outlier detection algorithms are also suitable. Only the features that showed the best separation of anomalous and normal samples in phase 1 are utilised at this stage.

To track the progress of the algorithm we employ supervised learning evaluation metrics. While if applied on fully labelled data these would be estimates of the algorithm's real-life performance, here they serve as a guide while tuning the algorithm. The metrics used in this work are precision and recall.

Newly discovered anomalies are reported as false positives, leading to a decrease in precision, while a stable high recall indicates that known anomalies are still detected. False positives are investigated manually and labelled (re-setting precision and recall) before repeating feature selection and training. This process stops when precision stabilises, i.e., no new anomalies are found.

It must be noted that this procedure tunes the algorithm to find known samples, and there is no corresponding way to obtain the precision-recall metric on new unlabelled data. To prevent overfitting of the training data, as well as to increase the explainability of the algorithm, feature engineering must be validated against domain expertise, and simpler and more interpretable features should be favoured. Without validation, the resulting features may be very specific to a particular piece of data for each anomaly in the training data but fail to generalize to unseen examples.

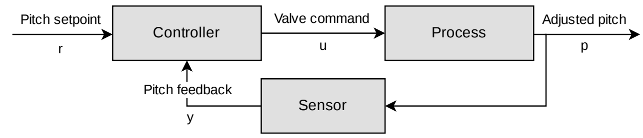

Fig. 2. The control scheme run by the PLC to control the pitch of the propeller of a thruster.

3. USE CASE: MARITIME MACHINERY

The presented framework is applied to the problem of detecting leakages and other undesired occurrences in maritime propulsion and manoeuvring systems, specifically, tunnel and azimuth thrusters. Manoeuvring systems can experience tough operating conditions that can lead to increased fuel consumption and wear of the equipment. While many deviations can be detected by simple rule-based algorithms operating on sensor data, others can present themselves in unpredictable ways and hence be difficult to discern from normal behaviour.

3.1 Maritime propulsion and manoeuvring systems

The thrusters are equipped with controllable-pitch propellers (CPP) regulated by a programmable logic controller (PLC) running a simple control scheme (Fig. 2). The controller uses the deviation between the pitch setpoint and the measured pitch angle to compute a voltage that adjusts a valve. Opening the valve allows for oil to flow into a cylinder, increasing pressure and pushing the propeller's pitch mechanism. There are known deadbands on the valve command voltage and pitch deviation.

When the valve is closed, the pitch should ideally be unchanging. In the presence of an internal leakage, some amount of oil still flows even if the valve is closed, which changes the pressure and the pitch angle.

3.2 Data

The available data consists of historical measurements from sensors installed on thrusters and information from control systems provided by Brunvoll, a maritime systems manufacturer. The data is gathered from four units located on two vessels in the years 2016 to 2019.

The pitch setpoint, valve command and measured pitch are used to construct the predictive features, while drive motor load is used as an indicator for whether the thrusters are running. The initial anomaly discovery phase utilizes additional variables, such as oil pressure and temperature, and thrust feedback and setpoint. These are not part of the final algorithm as they are not available for all thrusters.

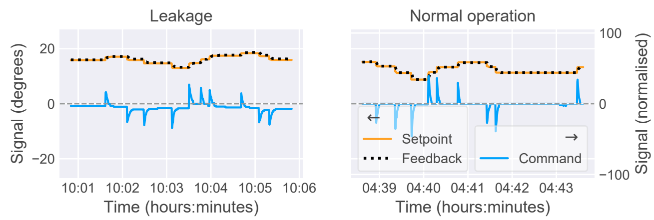

Fig. 3. Feedback follows the setpoint both in the presence of a leakage, and under normal operation. In the former, however, the command must compensate at all times, even when the setpoint is unchanging.

4. APPLICATION TO THE USE CASE

This section illustrates the application of the framework to the problem of condition monitoring in maritime machinery. Derived features, as well as data pre- and post-processing are specific to the use case.

4.1 Pre-processing

Periods where thrusters are active were divided into 5 min intervals, which resulted in ∼20 000 time-series samples. The time series were resampled at a rate of $\Delta t = 400$ ms, corresponding to the highest sensor frequency. Missing values were handled by forward filling. Valve command values within the control signal deadband were set to 0. Large, sudden jumps in the feedback are considered non-physical and assumed to be due to sensor errors. The time series displaying this behaviour were therefore removed from the training data.

4.2 Initial anomaly detection

The first anomalies were detected using the first two principal components of over 2000 features for pitch feedback, pitch setpoint and valve command extracted with *tsfresh*. In the next iterations, significant features were identified with the two-sample Kolmogorov-Smirnov test. We investigated 12 features with p-value < 0.0001: principal components derived from these features alone were more efficient in separating known anomalies from normal behaviour and led to the discovery of additional anomalies.

Leveraging domain knowledge, we formulated features that capture meaningful interactions between variables, such as the mean absolute deviation in pitch, and between estimated pitch and feedback. Additional anomalies were identified by investigating extreme values of summary statistics of these interaction-variables, e.g., mean and maximum values, variance, and standard deviation.

Observations with low servo pressure ($<$ 20 bar) and high drive motor load ($>$ 2000 kW) are clear outliers in the data, as illustrated in Fig. 4. During such time intervals, the pitch is deviating due to the pressure dropping without a corresponding command. Low drive motor load and low servo pressure are characteristic of starting of the motor and do not indicate an anomaly.

4.3 Informative features

Data exploration and the initial anomaly discovery in phase 1 resulted in four descriptive features that can detect

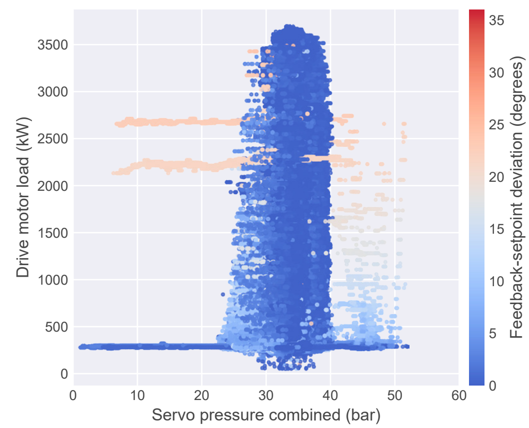

Fig. 4. Raw values of drive motor load and the sum of servo pressures. Low servo pressure with high drive motor load indicate an anomaly.

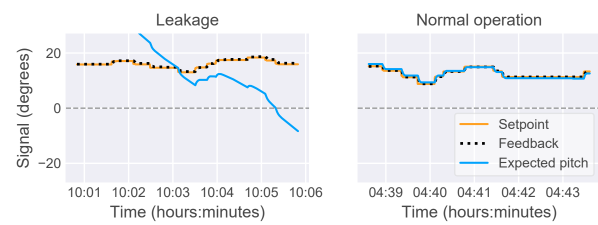

Fig. 5. In presence of a leakage (same as in Fig. 3), the measured pitch follows the desired setpoint, but the command given to maintain it is disproportionately large. Hence, the expected pitch from Eq. A.2 deviates from the measured pitch.

different types of physical anomalies, as well as sensor issues. Each 5 min interval is characterized by features x_1-x_4 based on the values of measured pitch feedback, pitch setpoint and valve command:

- x_1. Deviation of feedback from expected pitch: Pitch is estimated based on valve commands given and its initial value in each 5 min interval. Disproportionate commands indicate that the system has to compensate for some problems and show as a large deviation.
- x_2. The longest time with pitch deviation exceeding 4°: The control system reacts to pitch deviations above 4°, therefore not correcting for such deviation over time is indicative of problems with the machinery.
- x_3. Pitch feedback-setpoint deviation: High overall pitch deviation can indicate a lack of response or delayed response to the command.
- x_4. Maximal change in pitch feedback: Pitch feedback is a measured variable, and it is only feasible for it to change at a certain speed. Sudden, large jumps in the pitch feedback suggest sensor problems.

Detailed formulas for the derived features are in the Appendix.

4.4 Machine learning algorithm

Once the informative features have been identified, they can be used in a machine learning algorithm. In this work, we use a one-class support-vector machine (SVM), which is an unsupervised machine learning algorithm that bounds the feature space to represent one class (normal behaviour) (Tan et al., 2020). The new data which enters the model can be either predicted as this class or as an outlier, while the choice of the class boundary alters the tolerance to outliers.

4.5 Filtering out false positive samples

The algorithm returns not only anomalies but also some outliers that display abnormal, but not anomalous, behaviour. In the presented use case, some known abnormalities can be filtered out by simple rules based on the domain knowledge about thrusters.

Rapid setpoint changes: Feature x_1 is based on the assumption that the pitch can be described by a first order differential equation. However, it is observed that this assumption does not hold when the setpoint changes before a steady state is reached. To flag these false positives we consider the number of sign changes in the measured pitch error over some very short period of time (e.g., 3 timesteps, or 1.2 s). If the sign changes more than twice within this window, it indicates that the steady state has not been reached. Minimal feedback-setpoint deviations, e.g., $< 0.1°$, are ignored to adjust for the sensor noise.

Manual operation: In the manual operation mode, the control system is deactivated, so that the feedback change is not prompted by command and the setpoint remains stable. Manual mode can be detected if the pitch feedback changes over some period (e.g., 1 min) with no change in the command or setpoint.

5. RESULTS AND DISCUSSION

Since the anomalies are unknown in our use case, the performance of the algorithm cannot be quantified by standard evaluation metrics. We, therefore, present the algorithms application and ability to detect distinct types of anomalous behaviour, one of them being an internal leakage which was physically simulated in a testing setting.

5.1 Detected anomalies

The derived algorithm is able to detect anomalies that can be conceptualized into five categories:

- a_1. Compensating command: When a change in feedback requires unusually high command. Such occurrences suggest an internal leakage in a thruster. Detected by feature x_1.
- a_2. Delayed pitch response: When pitch reacts to the command but only after a certain delay. Detected by features x_1, x_3.
- a_3. Unreactive pitch: When the pitch does not react to the command. Detected by features x_1-x_3.
- a_4. Unreactive command: When the setpoint is changed and not reached by the feedback, but no new command is given. Detected by features x_2, x_3.
- a_5. Sensor issues: Characterized by very sudden changes in feedback or long constant periods in original features during activities in corresponding features. Detected by feature x_4.

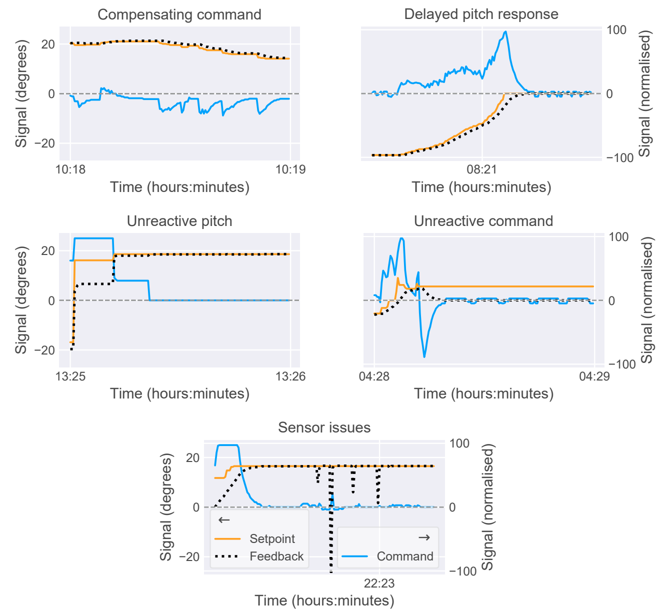

Fig. 6. Examples of anomaly types a_1-a_5.

Examples of the distinct anomaly types are presented in Fig. 6. The algorithm was also able to detect a physically simulated internal leakage in a thruster, which pertained to a_1 category. The comparison of this occurrence to the regular operation is presented in Fig. 3 and Fig. 5.

5.2 Classification of detected anomalies

Following the detection, the algorithm's interpretability can be increased by applying simple rules to categorize the anomalies, as described in the previous section.

It is important to note that purely rule-based detection, which potentially could be formulated after phase 1, would introduce the following trade-off: While strictly defined rules would limit the detection to specific anomalies, more general rules would result in an increase of false positives and lower the precision. We therefore argue that it is preferred to use an algorithm that defines desired behaviour and detects deviations from it, as proposed in this work.

5.3 Data quality

The algorithm is sensitive to the data quality and may falsely classify regular operation as anomalies if the provided sensor frequency is much lower than the one used in training, or in the presence of sensor noise. Alternative approaches, such as rule-based anomaly detection are however not resistant to these problems.

5.4 Application and deployment

In practice, the state of the thrusters is monitored over time, rather than on a single-occurrence basis. A systematic increase of anomalies in individual thrusters over time can be indicative of an actual problem, while a single anomaly may be treated as an unusual behaviour that is not an indication of problems, i.e., a false positive. This can be monitored as illustrated in Fig. 7. The probability of an

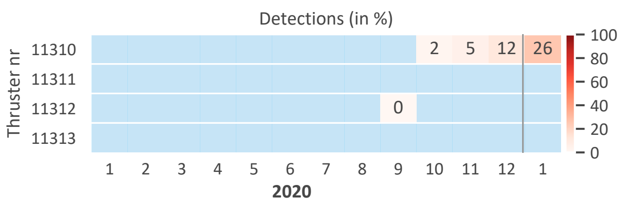

Fig. 7. Illustration of the proportion of anomalous to normal operation over time. The fictive thruster 11310 experiences a gradual increase of anomalies from Nov 2020 indicating wear. The thruster 11312 has $0 < x < 1\%$ anomalies in Sept 2020, which likely can be attributed to an unusual operation that is not a true anomaly.

actual anomaly can be evaluated based on a sliding window frequency of anomalous time series intervals relative to the total number of analysed time series in the period.

6. CONCLUSIONS

The proposed framework leads to a construction of an anomaly detection algorithm in the absence of labelled data and with initially unknown characteristics of anomalies. The framework fills the gap in the anomaly detection literature, where data-driven methods are called for, but neither purely supervised, unsupervised, nor deep learning algorithms are sufficient. The process focuses on deriving features that are indicative of anomalies, resulting in an interpretable algorithm. At the same time, expert feedback is used to ensure that the algorithm is not biased toward specific anomalies and potentially able to detect unseen anomaly types. The drawback of this approach is that since the features are derived to account for unseen anomalies, they may return some proportion of false positives.

The method is validated on a maritime propulsion and manoeuvring system, where five types of distinct anomalous states are identified. While the final algorithm returns two categories of false positives, they are easy to filter out with simple rules based on domain knowledge. The use case illustrates the entire process as well as the potential for deployment in real-time condition monitoring.

ACKNOWLEDGEMENTS

This work was funded by the Research Council of Norway via the MAROFF-2 project number 296465 and the data provided by Brunvoll AS.

REFERENCES

Beaugnon, A., Chifflier, P., and Bach, F. (2017). ILAB: An interactive labelling strategy for intrusion detection. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, 120–140. doi:10.1007/978-3-319-66332-6_6.

Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A. (2018). Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – a Python package). *Neurocomputing*, 307. doi:10.1016/j.neucom.2018.03.067.

Das, S., Wong, W.K., Dietterich, T., Fern, A., and Emmott, A. (2016). Incorporating expert feedback into active anomaly discovery. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 853–858. doi:10.1109/ICDM.2016.0102.

Dodge, Y. (2008). *Kolmogorov–Smirnov Test*, 283–287. Springer New York, New York, NY. doi:10.1007/978-0-387-32833-1_214.

Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. (2012). Toward supervised anomaly detection. *Journal of Artificial Intelligence Research (JAIR)*, 45. doi:10.1613/jair.3623.

Kim, D., Lee, S., and Lee, J. (2020). An ensemble-based approach to anomaly detection in marine engine sensor streams for efficient condition monitoring and analysis. *Sensors*, 20(24). doi:10.3390/s20247285.

Kong, X., Li, X., Zhou, Q., Hu, Z., and Shi, C. (2021). Attention recurrent autoencoder hybrid model for early fault diagnosis of rotating machinery. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–10. doi:10.1109/TIM.2021.3051948.

Pang, G., Shen, C., Cao, L., and Hengel, A.V.D. (2021). Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2). doi:10.1145/3439950. URL `https://doi.org/10.1145/3439950`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., and Louppe, G. (2012). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12.

Tan, Y., Tian, H., Jiang, R., Lin, Y., and Zhang, J. (2020). A comparative investigation of data-driven approaches based on one-class classifiers for condition monitoring of marine machinery system. *Ocean Engineering*, 201, 107174. doi:10.1016/j.oceaneng.2020.107174.

Vanem, E. and Brandsæter, A. (2019). Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine. *Journal of Marine Engineering & Technology*, 0(0), 1–18. doi:10.1080/20464177.2019.1633223. URL `https://doi.org/10.1080/20464177.2019.1633223`.

Zhang, Y., Li, X., Gao, L., Chen, W., and Li, P. (2019). Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method. *Measurement*, 151, 107232. doi:10.1016/j.measurement.2019.107232.

Appendix A. USE-CASE FEATURES

Features x_1-x_4 are based on the values of measured pitch feedback y, pitch setpoint r, valve command u and pitch deviation $e = y - r$. Hereafter, $y_i[t_j]$ denotes the pitch feedback for each 5 min interval i at time t_j. Likewise for r, u and e.

x_1: Deviation of feedback from expected pitch:

We assume that the pitch p is governed by the first-order linear system $\dot{p} = bu$, where b is an unknown constant input gain. Discretizing with sample time Δt and assuming zero-order hold, the pitch at time t_j can be expressed as:

$$p[t_j] = p[t_0] + b\Delta t \sum_{k=0}^{j-1} u[t_k]. \tag{A.1}$$

Hence, for time series i from time t_0 until t_n, and some estimate of b, denoted $\hat{b}$, the feedback and valve command can be used to calculate the expected pitch:

$$\hat{p}[t_j] = y[t_0] + \hat{b}\Delta t \sum_{k=0}^{j-1} u[k] \; \forall \, t_1 < t_j \leq t_n \tag{A.2}$$

Consequently, the constant b can be estimated for time series i by solving the least squares problem:

$$\hat{b}_i = \arg\min_b \|b \cdot \mathcal{U}_i - \mathcal{Y}_i\|_2 \,, \tag{A.3}$$

where $\mathcal{U}_i = \Delta t \left[u_i[t_0],\ u_i[t_0] + u_i[t_1],\ \ldots,\ \sum_{j=0}^{n-1} u_i[t_j]\right]$ and $\mathcal{Y}_i = [y_i[t_1] - y_i[t_0],\ y_i[t_2] - y_i[t_0],\ \ldots,\ y_i[t_n] - y_i[t_0]]$. Here, $\hat{b}$ is computed by taking the average of $\hat{b}_i$ for a set of time series without known anomalies.

The deviation of feedback from expected pitch is defined as the mean absolute deviation of the expected and measured pitch for time series i:

$$x_{i1} = \frac{1}{n} \sum_{j=1}^{n} |\hat{p}_i[t_j] - y_i[t_j]| \tag{A.4}$$

x_2: The longest time with pitch deviation exceeding $4°$:

The maximal number of consecutive timesteps where the pitch deviation exceeds 4°is expressed as:

$$x_{i2} = \max_{q,r} \sum_{j=q}^{r} \mathrm{sign}(e_i[t_j] - 4), \tag{A.5}$$

such that $\mathrm{sign}(e_i[t_j]-4) = 1$ and $q \leq r, q, r \in \{0, 1, \ldots, n\}$.

x_3: Pitch feedback-setpoint deviation:

$$x_{i3} = \frac{1}{n} \sum_{j=0}^{n} |e_i[t_j]| \tag{A.6}$$

x_4: Maximal change in pitch feedback:

$$x_{i4} = \max(|y_i[t_j] - y_i[t_{j+1}]|), \, \forall \, j \in \{0, n-1\} \tag{A.7}$$