Detecting the Onset of Machine Failure Using Anomaly Detection Methods [⋆]

Mohammad Riazi[1], Osmar Zaiane[1], Tomoharu Takeuchi[3], Anthony Maltais[2], Johannes Günther[1], and Micheal Lipsett[2]

[1] University of Alberta, Dep. of Computing Science Edmonton AB, CANADA
[2] University of Alberta, Dep. of Mechanical Engineering, Edmonton AB, CANADA
[3] Mitsubishi Electric Co., Information Technology R&D Center, Kanagawa, JAPAN
{riazi, zaiane}@ualberta.ca

Abstract. During the lifetime of any machine, components will at some point break down and fail due to wear and tear. In this paper we propose a data-driven approach to anomaly detection for early detection of faults for a condition-based maintenance. For the purpose of this study, a belt-driven single degree of freedom robot arm is designed. The robot arm is conditioned on the torque required to move the arm forward and backward, simulating a door opening and closing operation. Typical failures for this system are identified and simulated. Several semi-supervised algorithms are evaluated and compared in terms of their classification performance. We furthermore compare the needed time to train and test each model and their required memory usage. Our results show that the majority of the tested algorithms can achieve a F1-score of more than 0.9. Successfully detecting failures as they begin to occur promises to address key issues in maintenance like safety and cost effectiveness.

Keywords: Anomaly detection · Fault detection · Predictive Maintenance · Machinery Diagnostics · Onset of failure · Machine learning.

1 Introduction

Numerous factors can contribute to the quality of a product, and not every one of these factors are under manufacturers control. One of the most common sources of quality problems is *faulty equipment* which has not been properly maintained [10]. Hence, monitoring the condition of machines and components such as cooling fans, bearings, turbines, gears, belts and maintaining a desirable working state becomes very crucial.

When a machine or a component fails, corrective maintenance is performed to identify the cause of failure and decide on repair procedures required to maintain and re-initiate the machine to its normal working conditions. However, because the machine has already failed without any prior warnings, time is required for

[⋆] Funding support is gratefully acknowledged from the Natural Sciences and Engineering Research Council of Canada, the Alberta Machine Intelligence Institute and Mitsubishi Electric Corporation, Japan.

procuring and repairing of the failed component. Therefore, a maintenance strategy needs to be considered to minimize the downtime of a service. But machines and their components degrade through time, and the time of failure is not known in advance. Hence, time-based maintenance strategies are predominantly used for maintaining the conditions of machines and equipment [11].

There are many advantages to using scheduled maintenance. First, due to regular checks, the operational reliability of the machine is extended and catastrophic failures are somewhat prevented. Second, the time required to maintain the equipment is drastically reduced because the necessary tools and resources are obtained well in advanced. But, there still remains substantial disadvantages including, costs pertaining to tools and experts performing regular maintenance even-though it might not be necessary at all. Another problem is unforeseen failure that might occur during maintenance intervals. Moreover, during a regular maintenance a part may end up being replaced regardless of its considerable Remaining Useful Life (RUL), which incurs additional costs.Last but not least, the risk of failure may increase during maintenance due to improper diagnosis and or because of intrusive measures taken during repairs.

Prognostics and Health Management (PHM).is a discipline that consists of technologies and methods to evaluate the reliability of a product in its actual life cycle conditions to detect the development of failure and mitigate system risk [3]. Sensor systems are needed for PHM for online monitoring of an equipment. With online monitoring the condition of the equipment is constantly checked throughout its life cycle to identify any potential failure. This strategy is referred to as Condition-Based Maintenance (CBM). PHM can be implemented in several ways: physics-based models, knowledge-based models and data-driven models.

During recent years, data-driven models also known as data mining methods or machine learning methods for machine health monitoring are becoming more attractive due to the substantial development of sensors and computing platforms. Data-driven methods use historical data to automatically learn a model of system behaviour. Features that encompass the health state of a machine are extracted using a series of procedures. These features are then used by a variety of machine learning methods to decide on the health state of the machine. Based on the availability of datasupervised, semi-supervised, unsupervised or reinforcement learning could be used to achieve the desired result.

Unfortunately, there are no freely published datasets available on machine degradation through time. The purpose of this study is two fold: Design and development of a robot-arm platform that can generate data at different operational states; Apply and compare a number of semi-supervised anomaly detection techniques on the data collected for detecting the ***onset*** of failure.

2 Background

Fault detection is a major task across many applications [1] including quality control, finance, security and medical. In an industrial or manufacturing settings, an equipment or product fault is an abnormal behaviour that may lead to the total failure of a system. In general, there is a time gap between the appearance of a fault and its ultimate failure. This time gap can be used by condition-based maintenance to reduce the probability of failure by monitoring the stages of failure and prevent fault from becoming a failure.

As mentioned previously, PHM can be implemented using different approaches: physics-based models based on mathematical models [7], knowledge-based models formalized using if-then rules, and data-driven-based models which mainly rely on anomaly detection from data.

2.1 Anomaly detection methods

There exists numerous anomaly detection methods. A rough grouping of some of these methods include, statistical algorithms, clustering-based, nearest neighbour-based, classification-based, spectral-based, space subsampling-based and deep learning methods.

Statistical-based methods assume that the data follows a specific distribution, so the model is created from a specified probability distribution [18]. The simplest approach for detecting anomalies in data would be flagging data points that deviate from common statistical properties of a distribution.For example, one could define an anomaly based on a certain standard deviation away from the mean [12]. The advantage of these models is that they output a probability as a measure of outlierness.

In **clustering-based anomaly detection**, the assumption is that data points that are similar belong to a similar group. This is determined by the distance to the cluster centroid. The anomaly score is then calculated by setting a threshold for the size of a cluster or the distance to the cluster centroid; if cluster has data points less than the value of threshold they are marked as anomalies, or if the data points distance to the centre of the cluster exceeds the set threshold it is flagged as anomalous. Self-Organizing Maps (SOM), k-means are examples of such methods.

The **nearest-neighbour-based anomaly detection** generally assumes that normal data samples appear in neighbourhoods that seem to be dense, while anomalies are far from their closest neighbours. Nearest neighbour methods can be generally grouped into distance-based and density-based methods. Both approaches require a similarity or a distance measure in order to make the decision on the degree of abnormality of a data instance. Some examples include, k-NN, Mahalanobis distance, and Local Outlier Factor (LOF).

Classification-based anomaly detection can be divided into one-class (normal labels only) and multi-class (multiple classes) classification depending on the availability of labels. Anomaly detection based on a classifier comprises of two steps [2]: The first step, which is the training phase, a classifier is learned using available labeled training data. In a second step, the test instances are classified as normal or abnormal using the classifier trained in initial step. The One-Class Support Vector Machine (OCSVM) and neural network methods are examples of such detection methods.

Spectral or subspace-based methods try to extract features that best describe the variability of the training data [2]. These methods assume that the normal data can be presented in a lower dimension subspace where normal data is distinguished from abnormal data. Principle component analysis is considered as a subspace-based approach to anomaly detection.

Many outlier detection methods methods suffer from curse of dimensionality; as the dimensionality of a given dataset increases, distance based approaches fail because the relative distance between any pair of points become relatively the same [5]. To overcome this problem **sub-sampling-based methods** the high dimensional space is divided into smaller sub-spaces and the outlier ranking in different sub-spaces is monitored. Outliers are points that consistently rank high in the smaller sub-spaces. The T*-Framework [5], and Isolation forest [8] are algorithms that divide the high dimensional space to smaller dimensions and try to find anomalies in the lower dimensional space.

Deep learning is a descendant of machine learning that tries to model high level representations hidden in a dataset and classify or predict patterns by stacking multiple layers of neurons or processing modules in a hierarchical structure. Each layer learns a new representation of the input data and then by stacking multiple layers, complex patterns are learned. This eliminates expert knowledge and designing hand crafted features, thus the models can be applied to machine health monitoring in a very general way [16].

3 Experiment Design

A belt-driven, single degree of freedom, re-configurable robot manipulator has been customized for this investigation. The system was developed to simulate faults that can occur during regular operation of the arm and to assess the effect of mass, friction and belt tension on motion. Multiple reproducible experiments could repeatedly be run on it with similar results.

3.1 The Robot arm Platform

The robot arm is designed to move in a back and forth manner and following a predefined trajectory. Even though this seems like a simple machine, there exist many real life examples such as opening and closing gates in elevator doors that require fault monitoring due to constant use and importance of up-time for providing service.

The robot arm consists of many components that help with its operation, data acquisition and reproducibility. A picture of the complete robot arm platform is demonstrated in Figure 1.

The system is controlled by a programmed teensy 3.2 microcontroller. It is actuated by a brushless stepper motor to drive the belt through a programmed range of motion with a Leadshine DM542 Microstep Drive. This range of motion is measured by two US Digital E2 optical encoders. The driving belt is adjustable in tension. To measure the instantaneous tension in the belt, there are two strain gauges. As the tension has an influence on the systems functioning, its value is needed to reset the belt tension back to its normal operating value. Another influence on the system, the ambient temperature is measured by a thermo couple. A third influence is the weights that are used to change the inertia. For most experiments, 5.5 kg weights are placed on the arm. Lastly, to minimize external vibrations the arm is connected to a 3-ton seismic mass.

3.2 Data acquisition

The arm is armed with four sensors: two full-bridge strain gauge configurations and two encoders. For the purpose of this experiment, only the encoder on the motor output shaft is used because we are interested in measuring the required torque for moving the arm. The data acquisition system outputs 7 values through the USB ports connected to a laptop. These measurements are: Timestamp, cycle number, cycle mode (forward or backward), Motor shaft angle (in degress), Motor output torque (Nm), Belt tension (N), and ambient temperature (degree cel.) around the arm. A representative set of complete cycles of the torque under normal belt tension set at 150 N is shown in red in Figure 2.

3.3 Failure Types

After establishing a baseline normal torque profile, a number of typical faults expected in a similar system were simulated. In order to account reproducibility, multiple sensors were installed on the robot arm. These are shown in figures 1(a) and 1(b). In total, five different fault modes were simulated: The belt was loosened on two different to either (1) 120N or (2) 100N. For the third experiment the belt was tightened to (3) 168N. Sand was scattered on the surface for the fourth experiment. The fifth experiment simulated a increase in the ambient temperature to 40 degrees Celcsius.

Table 1 summarizes the data collected from the robot arm platform. To have a better understanding of the tensions, Figure 2 shows the trend of tension over time for the normal state and the interval at which each fault falls considering the tension value. Moreover, to show how close and subtle the faults are in relation with the normal profile, overlaid torque signals for normal and fault conditions are also shown.

3.4 Data Pre-processing

A majority of the outlier detection algorithms use distance (Euclidean) as a measure of similarity and assume normality of input data. Hence feature normalization is required to approximately level the ranges of the features and enforce the same effect in the computation of similarity [13]. For the purpose of this study, torque samples were rescaled so the features have the properties of a standard normal distribution with mean zero and a standard deviation of one.

Torque data is collected through a data acquisition system which is guaranteed to output reliable measurements in a structured format. The torque is

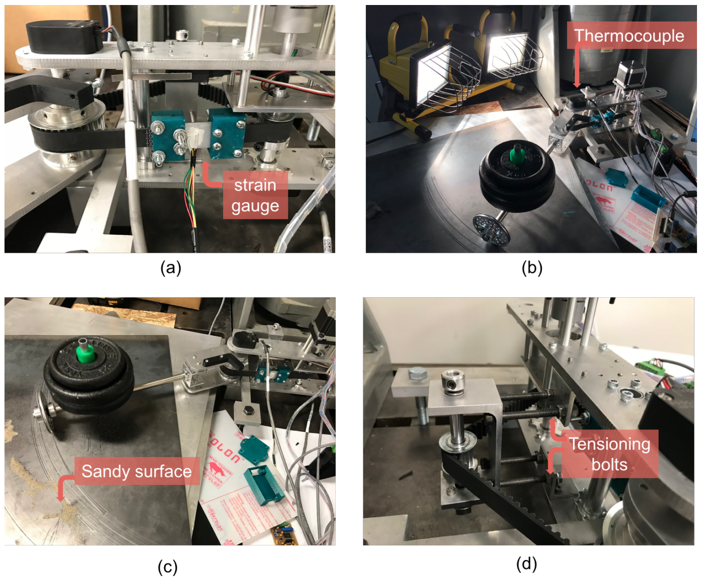


Fig. 1. Sensors installed on the belt and the arm and simulated faults. (a) Strain gauge for measuring the belt tension. (b) Thermocouple for measuring ambient temperature with halogen bulbs for generating heat to see the effect of temperature on the belt tension. (c) Friction using sandy surface. (d) Loose and tight belt tension faults using loosening and tightening of bolts and adjusting the height of idler pulley.

	TEMP	TENSION	NO. SAMPLES
Normal	˜23.5	˜150	7193 (˜22 hrs)
Loose L1	˜23.5	˜120	182
Loose L2	˜23.5	˜99-100	180
Tight	˜23.5	˜168	194
High Temperature	˜40-42	˜166-169	210
Sand (Friction)	˜23.5	˜150	183

Table 1. Summary of data collected from the robot arm platform

sampled at 200Hz and each observation should contain 2000 sampled data points per cycle. The collected samples are saved in a tabular CSV file with each row representing a single observation with 2000 sampled data points.

3.5 Feature-based representation of Torque Waveforms

Feature-based representation of time series have been used across many disciplines ,e.g. medical or speech recordings [6]. Constructing these features is usually subjective for a given dataset.A very simple example of representing a time series is using its mean and variance characteristics.

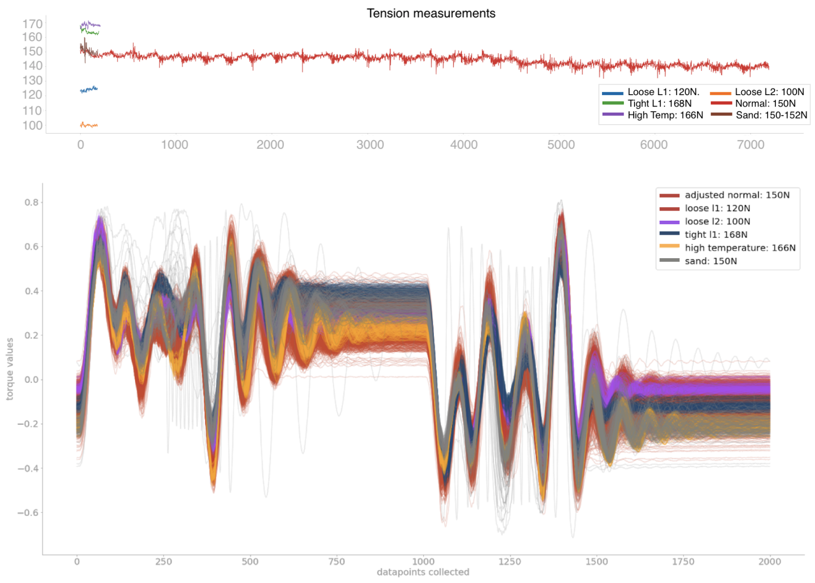


Fig. 2. Top: Normal vs faulty tension signals demonstrated through time —**Bottom**: Overlaid torque signals for normal and faulty data

As a first attempt to construct features from the torque dataset (2000 data points per sample), we followed the work of [9] and used the first order statistics along other descriptive measures to extract a total of 11 features.

In a second attempt, additional features were extracted using the "Time Series Feature extraction based on scalable hypothesis tests" abbreviated as tsfresh python package [4]. The package contains numerous feature extraction methods and a robust feature selection algorithm. Using the tsfresh library, a total of 714 features were extracted, which was further pruned to 114 using the built-in feature selection method.

To further prune the extracted features and select the most important and discriminating features, we have used the gradient boosting machine implemented in LightGBM library and the Pearson correlation to remove the low importance and correlated features in both manually extracted feature sets and the tsfresh generated features.

The final set of features were reduced from 11 to 8, including mean, variance, skewness, kurtosis, mean, variance, skewness, kurtosis, and RMS for the manual features. From the automatic tsfresh features, 76 out of 114 were chosen.

3.6 Feature learning & extraction using auto-encoders

We use a 1D segment from raw input data (torque data) as the input to our CNN auto-encoder because the torque signal is correlated only in the time domain, this is followed by 1D filters in the convolutional layers of the model.

Evidently, a deep learning model with deeper architecture achieves a better performance compared to a shallow one. However, a deeper network results in a more complicated number of hyper-parameters and a variety of network architectures. This in fact makes the building of an appropriate and efficient model quite difficult [14]. For the purpose of this study we have particularly followed the work of Zhang et al., [15] to determine the 1D-CNNAE architecture.

The proposed 1D-CNNAE consists of a CNN encoder network and a mirrored decoder network with 5 layers of alternating convolution and pooling layers. The first convolutional layer consists of 64 filters (dimensionality of output space) and kernel size of 1, followed by a max-pooling layer. The subsequent layers are four alternating small (10x1) convolutional layers followed by max-pooling.

The learned features from the bottleneck layer of the CNNAE network is extracted and saved to use with the anomaly detection methods to further investigate the performance of the automatically extracted features.

4 Training Procedure

The normal dataset was split into train, validation and test sets with a ratio of 80%, 10%, and 10%, respectively. The train and validation datasets are used at the time of training with the validation dataset used for early stopping criterion for the auto-encoder to prevent overfitting. The normal *test* dataset was set aside to be merged with the abnormal test datasets to obtain a more realistic testing set. The algorithms were first trained using the normal only dataset with the raw torque values (2000 features), the manually extracted features, tsfresh features and the features extracted by CNNAE. A contamination ratio which determines the amount of contamination of the data set, i.e., the proportion of outliers in the data set was predetermined (once assumed 90% of data are normal and once 95%) and used when fitting each model to define the threshold on the decision function. Once the detectors were fit, the outlier scores of the training set is determined using the corresponding decision function of a detector.

The nearest rank method is then used to find the ordinal rank of the anomaly scores outputted by the detectors.

5 Results and Concluding Remarks

The trained models were used on the unseen mixed normal and anomalous dataset to determine the anomaly scores. The anomaly scores were then categorized into anomalous and normal according to the set threshold (assuming 90% & 95% normal data). An anomaly score beyond the threshold is then flagged with label 1 as being anomalous and 0 if smaller than threshold.

Figure 4 shows a side-by-side comparison of performance of each algorithm using the 8 manual features. The results show that the performance of all anomaly detectors using the hand crafted features is comparable to that of the raw torque measurements/features. Similarly, the 76 tsfresh features also produce comparable results to that of the 8 manually created and raw features, showing that the additional features extracted have no or near to none performance increase in the models. We recommend the isolation forest anomaly detection if a

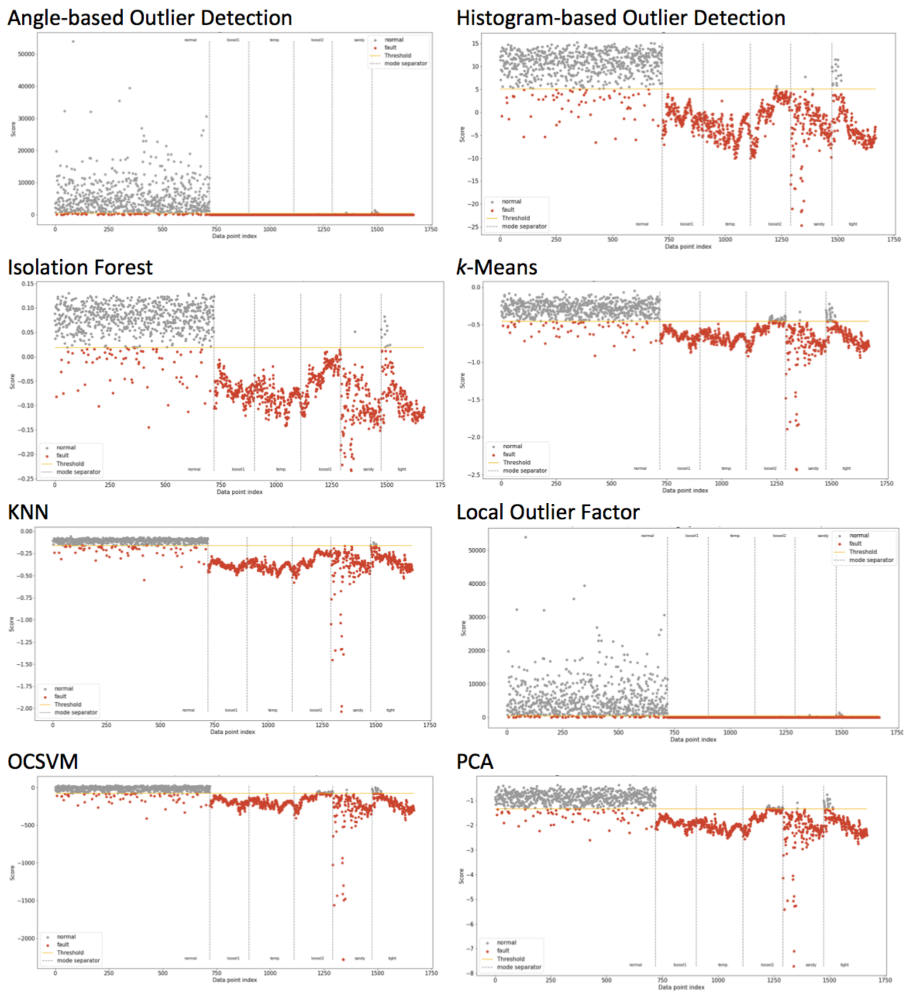
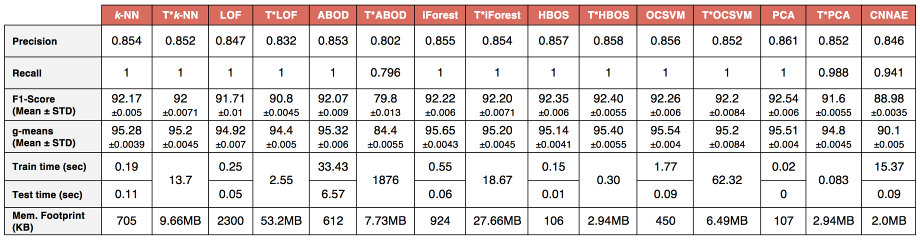


Fig. 3. Trained models applied to mixed data (normal + fault modes) with threshold set at 90%. Grey points mark normal, orange points mark abnormal data.

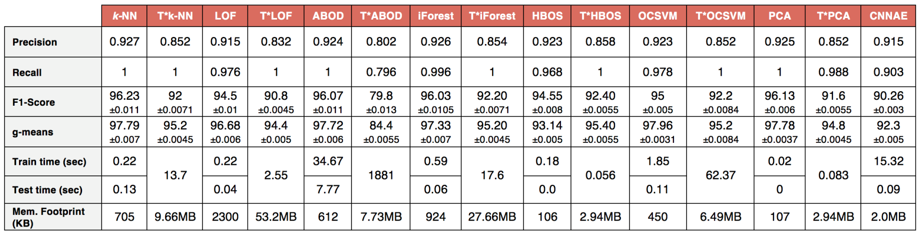
single algorithm is to be choosen. Although based on the evaluation metrics, the algorithm is slightly slower in training time, some of its properties makes it the winner among other algorithms. These properties include: no need for calculating distance or density of samples; at the time of training a tree is constructed and at test time it passes sample points down the tree to calculate the average number of edges required to reach an external node. It is very robust in terms of choosing parameters; to configure it, one would need to (1) provide maximum samples to draw from training dataset to train each of the base estimators, (2) number of base estimators (the number of ensembles), (3) contamination, the ratio of outliers in the dataset (if used as a semi-supervised algorithm). Liu et al., [8], show that the choice of sample size is very robust and that the algorithm converges very fast with low sample size. We think the most important configuration parameter is the contamination ratio, which stands the same for all the other algorithms. However, according to Liu et al., the choice of contamination may be rectified by using a larger sampling size. In addition to its performance, the memory requirement for isolation forest is very low because of sub-sampling and so it makes it a great practical choice for deployment and in industry.

Figure 3 illustrates the anomalous scores of normal and faulty signals on a scatter plot with grey marking points identified as normal and orange points as anomalous for manually constructed features. The threshold (90%) is represented as a golden horizontal line, marking the cut-off at which a sample is flagged normal or anomalous.



	k-NN	T**k*-NN	LOF	T*LOF	ABOD	T*ABOD	iForest	T*iForest	HBOS	T*HBOS	OCSVM	T*OCSVM	PCA	T*PCA	CNNAE
Precision	0.854	0.852	0.847	0.832	0.853	0.802	0.855	0.854	0.857	0.858	0.856	0.852	0.861	0.852	0.846
Recall	1	1	1	1	1	0.796	1	1	1	1	1	1	1	0.988	0.941
F1-Score (Mean ± STD)	92.17 ±0.005	92 ±0.0071	91.71 ±0.01	90.8 ±0.0045	92.07 ±0.009	79.8 ±0.013	92.22 ±0.006	92.20 ±0.0071	92.35 ±0.006	92.40 ±0.0055	92.26 ±0.006	92.2 ±0.0084	92.54 ±0.006	91.6 ±0.0055	88.98 ±0.0035
g-means (Mean ± STD)	95.28 ±0.0039	95.2 ±0.0045	94.92 ±0.007	94.4 ±0.005	95.32 ±0.006	84.4 ±0.0055	95.65 ±0.0043	95.20 ±0.0045	95.14 ±0.0041	95.40 ±0.0055	95.54 ±0.004	95.2 ±0.0084	95.51 ±0.004	94.8 ±0.0045	90.1 ±0.005
Train time (sec)	0.19	13.7	0.25	2.55	33.43	1876	0.55	18.67	0.15	0.30	1.77	62.32	0.02	0.083	15.37
Test time (sec)	0.11		0.05		6.57		0.06		0.01		0.09		0		0.09
Mem. Footprint (KB)	705	9.66MB	2300	53.2MB	612	7.73MB	924	27.66MB	106	2.94MB	450	6.49MB	107	2.94MB	2.0MB

(a)



	k-NN	T*k-NN	LOF	T*LOF	ABOD	T*ABOD	iForest	T*iForest	HBOS	T*HBOS	OCSVM	T*OCSVM	PCA	T*PCA	CNNAE
Precision	0.927	0.852	0.915	0.832	0.924	0.802	0.926	0.854	0.923	0.858	0.923	0.852	0.925	0.852	0.915
Recall	1	1	0.976	1	1	0.796	0.996	1	0.968	1	0.978	1	1	0.988	0.903
F1-Score	96.23 ±0.011	92 ±0.0071	94.5 ±0.01	90.8 ±0.0045	96.07 ±0.011	79.8 ±0.013	96.03 ±0.0105	92.20 ±0.0071	94.55 ±0.008	92.40 ±0.0055	95 ±0.005	92.2 ±0.0084	96.13 ±0.006	91.6 ±0.0055	90.26 ±0.003
g-means	97.79 ±0.007	95.2 ±0.0045	96.68 ±0.006	94.4 ±0.005	97.72 ±0.006	84.4 ±0.0055	97.33 ±0.007	95.20 ±0.0045	93.14 ±0.005	95.40 ±0.0055	97.96 ±0.0031	95.2 ±0.0084	97.78 ±0.0037	94.8 ±0.0045	92.3 ±0.005
Train time (sec)	0.22	13.7	0.22	2.55	34.67	1881	0.59	17.6	0.18	0.056	1.85	62.37	0.02	0.083	15.32
Test time (sec)	0.13		0.04		7.77		0.06		0.0		0.11		0		0.09
Mem. Footprint (KB)	705	9.66MB	2300	53.2MB	612	7.73MB	924	27.66MB	106	2.94MB	450	6.49MB	107	2.94MB	2.0MB

(b)

Fig. 4. A side by side comparison of the anomaly detection algorithms applied to mixed data (normal + fault modes) with threshold set at (a) 90% and (b) at 95%.

6 Future Work

In a future study, a comparative assessment of the presented anomaly detection techniques will be made against the physics-based models of the normal and the different fault modes (loose belt, tight belt and friction).

References

[1] Charu C Aggarwal. “Outlier analysis”. In: *Data mining*. Springer. 2015, pp. 237–263.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: a survey”. In: *ACM computing surveys* 41.3 (2009), p. 15.

[3] Shunfeng Cheng, Michael H Azarian, and Michael G Pecht. “Sensor systems for prognostics and health management”. In: *Sensors* 10.6 (2010), pp. 5774–5797.

[4] Maximilian Christ et al. “Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)”. In: *Neurocomputing* (2018).

[5] Andrew Foss, Osmar R Zaïane, and Sandra Zilles. “Unsupervised class separation of multivariate data through cumulative variance-based ranking”. In: *Intl. Conf. on Data Mining (IEEE ICDM)*. 2009, pp. 139–148.

[6] Ben D Fulcher, Max A Little, and Nick S Jones. “Highly comparative time-series analysis: the empirical structure of time series and their methods”. In: *Journal of the Royal Society Interface* 10.83 (2013), p. 20130048.

[7] Rolf Isermann. “Model-based fault-detection and diagnosis–status and applications”. In: *Annual Reviews in Control* 29.1 (2005), pp. 71–85.

[8] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *Intl. Conf. on Data Mining (IEEE ICDM)*. 2008, pp. 413–422.

[9] Alex Nanopoulos, Rob Alcock, and Yannis Manolopoulos. “Feature-based classification of time-series data”. In: *International Journal of Computer Research* 10.3 (2001), pp. 49–61.

[10] Reuters Editorial News. *4 Ways Predictive Maintenance Streamlines Manufacturing*. 2017. URL: `http://www.reuters.com/media-campaign/brandfeatures/intel/gated/TCM_1610337_intel_adlink_IOT_whitepaper_R6.pdf`.

[11] Michael Pecht. “Prognostics and health management of electronics”. In: *Encyclopedia of Structural Health Monitoring* (2009).

[12] Mahesh S Raisinghani et al. “Six sigma: concepts, tools, and applications”. In: *Industrial management & Data systems* 105.4 (2005), pp. 491–505.

[13] Mostafa A Salama, Aboul Ella Hassanien, and Aly A Fahmy. “Reducing the influence of normalization on data classification”. In: *Proceedings of the International Conference on Computer Information Systems and Industrial Management Applications*. IEEE. 2010, pp. 609–613.

[14] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical bayesian optimization of machine learning algorithms”. In: *Proceedings of the International Conference on Advances in neural information processing systems*. 2012, pp. 2951–2959.

[15] Wei Zhang et al. “A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals”. In: *Sensors* 17.2 (2017), p. 425.

[16] Rui Zhao et al. “Deep learning and its applications to machine health monitoring: a survey”. In: *arXiv preprint arXiv:1612.07640* (2016).