National Research University Higher School of Economics
Faculty of Computer Science
Programme 'Master of Data Science'

MASTER'S THESIS
topic

'Application of Anomaly Detection Methods in Industrial Time Series'

Student: Filatova Veronika Nikolaevna

Supervisor: Chankin Andrey Andreevich, Pasha Bank, Expert Data Scientist

Moscow, 2024

ABSTRACT

The thesis explores the application of anomaly detection methods in multi-dimensional industrial time series on the example of labelled data from sensors of the water circulation system. The data includes sensor readings collected during more than 30 experiments. During each experiment, the normal operation of the system was replaced by an abnormal stage due to the unstable operation of the system.

The main objective of the work is to apply machine learning and deep learning models to identify their effectiveness in the task of system failures detection.

Keywords: time series, industry, multi-dimensional time series, anomaly detection, machine learning.

TABLE OF CONTENTS

ABBREVIATIONS

AE – Autoencoder;
ARMA – Autoregressive Moving Average;
CNN – Convolutional Neural Network;
DNN – Deep Neural Network;
KNN – K-Nearest Neighbour;
LSTM – Long Short-Term Memory Network;
OCSVM – One-Class Support Vector Machine;
RNN – Recurrent Neural Network;
SKAB – Skoltech Anomaly Benchmark;
TN – true negatives;
TP – true positives;
FN – false negatives;
FP – false positives.

INTRODUCTION

High-quality performance of anomaly detection tasks plays tremendously important role in the industrial domain, since the inability to identify anomalies makes problems such as equipment failure unpredictable and not always explicable.

At the same time, industrial data often changes over time. Accordingly, when studying indicators, it is necessary to take into account the time factor, methods of detecting anomalies for ordinary data may not be effective. What is more, complex systems are usually used for the implementation of various processes in industry, the operation of which depends on many factors. Therefore, it should be possible to detect anomalies is the multidimensional data.

The application of machine learning methods to identify anomalies in industrial multidimensional time series is analysed in this work. The object of the study is the time series from the the Skoltech Anomaly Benchmark (SKAB) [11]. The data were obtained in real laboratory conditions by conducting an experiment. The water circulation system is designed to simulate a water supply system in laboratory conditions and ensure water circulation through pipes using a pump. There were failures in the operation of the system, which were registered.

The purpose of the work is to evaluate the effectiveness of using machine learning methods to detect system failures. To achieve this goal, the following tasks have been set:

1) to study the existing approaches and methods for detecting anomalies in multidimensional time series;
2) to analyse analyse the considered industrial data;
3) to apply anomaly detection methods for system malfunctions detection.
4) to evaluate the effectiveness of the selected methods.

1 THEORETICAL ASPECTS OF ANOMALY DETECTION IN INDUSTRIAL TIME SERIES

1.1 Characteristics of industrial time series

A time series is an ordered sequence of points or a feature measured at constant time intervals, representing a characteristic of a process. Time series data is characterized by dependence between points of time and one or more features describing the defined phenomenon. The time-stamp in such data can not be ignored as it gives a significantly important information, and it can not be treated as one of dimensions. This is why data ordered by time requires special tools for anomaly detection.

The importance of time series analysis can not be overestimated due to the specific characteristics of such a data. Firstly, time can be considered as a special variable, which moves only in one direction. Secondly, time is an input which is taken by processes – time itself can influence on the result of the process. Thirdly, problems of a business involve a time component often [1]. For example, it is important for a company to provide the growth of a profit during some periods of time or forecast demand for manufactured products in the future period.

There are some crucial characteristics of time series data:

- autocorrelation: the value of data point at time T-1 directly affects its data point at time T;
- smooth movement: significant changes in data take some time usually, this is why we do not expect sudden fluctuations in the ordinary data and if they happen, they can be considered as anomalies;
- the nature of change: trend, seasonality, cyclical variation, noise [1].

Trend is a characteristic of data changes which shows the general direction of the movement: upward or download. Seasonality means that there can be periodic patterns in the changes of data. Cyclical variation is close to the definition of seasonality, but it is not as recurring and stable. Noise implies some randomness of the data and should be very small percentage of the data (close to 0), otherwise it could be a signal that some important information about dataset was not examined.

What is more, when we deal with time series data, it is significantly important to have dataset big enough for defining patterns in the data changes and understand reasons for these changes. Time series data can be univariate (consists of one feature) or multi-variate (includes few features).

There are many application areas for time series analysis: economics, retail, banking, cybersecurity, industry, medicine, etc. The following tasks are set on the time series: modelling, forecasting, feature extraction, classification, clustering, pattern detection, anomaly detection. For example, in the industrial domain sequences of observations could be collected from multiple sensors of the equipment, critical deviation could be considered as anomaly since it indicates a malfunction or some crucial changes in the process.

Anomaly detection – is a task to discover an unusual behaviour for some of the observations. In addition to this, reasons for an unexpected behaviour should be analysed. For instance, outliers could be related to errors or noise in data, in this case it is better to clean the data. On the other hand, outliers could be a signal of important changes in the process and give meaningful information. The example of such situation is a fraud detection or equipment failure.

The problem of anomaly detection is a problem of defining which time series is significantly different from others in a set of multiple time series. Suppose, there is a dataset D (1).

$$D = \{X_i | i = 1, 2, ..., m\}, \tag{1}$$

where $X_i = \{x_i(t) | 1 \le t \le n\}$,

$x_i(t)$ – the value of the ith time series at time t,

n – the length of the time series,

m – the number of time series in the dataset [3].

Then we need to estimate outlierness $O(X_i)$ and compare it with the defined threshold. If the outlierness is bigger or equal to threshold, then series is considered to be anomalous (2).

$$O(X_i) \ \ge \ \theta_o, \tag{2}$$

where $O(X_i)$ – outlierness or anomaly score,

$x_i(t)$ – the value of the ith time series at time t,

θ_o – threshold [3].

Types of the anomalies in time series data can be divided into two groups: point and collective. The point (local) anomaly represents a data point, whose noise constituent is extremely (unexpectedly) large. The collective anomalies represent behavioural changes over time: the trend after the first outlier differs from the trend before, this first outlier is a change point [1].

Time series data values can not be considered as independent points, since the next one point is highly influenced by the previous one. As a result, context plays a critical role and change detection is related to the anomaly detection problem in time series.

There two ways for change's occurring in the data: concept drift and abrupt change. The first type is connected with slow changes over time and can be detected with an analysis over long time. The second type happens in situations, when "the underlying data mechanism has somehow changed fundamentally"[2].

It is important to define change-point detection as a separate task due to the fact, that data for this task is often unsupervised or semi-supervised (so it is not possible to consider the task as binary classification), other metrics are used (for example we could be interested not in the amount of outliers, but in the delay time). Change-point detection can be online or off-line. In the first category the most important purpose is to define anomalies optimally, in the second category - as early as possible.

1.2 Anomaly detection methods used for time series

The task of finding anomalies during the technological process is the most important link in the cycle of technical diagnostics. The solution to this problem ensures the safety, efficiency and reliability of the technological process. The assumption of temporal continuity is an important feature of such a process: abrupt changes are not expected to happen if there are not unusual behaviour at work. But this assumption is more relevant for univariate than multidimensional data [2].

Since the level of temporal continuity is significantly different for univariate and multivariate time series data, different methods are used to determine anomalies in each case. Applicable methods for multidimensional time series are more similar to traditional multivariate outlier detection, the main difference is in the temporal component. There are two main kinds of the outliers: an outlier in the individual recordings and alterations in the aggregated trends. The second one involves the aggregated changes on a particular time window and often begins with point outlier – change point [2].

The are different approaches to the classification of anomaly detection methods. One of the classification methods is based on the presence of labels in the analysed data. According to this approach, there are supervised, semi-supervised and unsupervised methods. [10]

Supervised methods are based on the use of datasets, labelled by domain experts. In this case every observation is marked as normal or abnormal and outlier detection can be considered as classification problem. Then the task for a model is to define outliers and normal observations. The dataset for this task should be divided into training and test samples. However, there are a number of problems in using this approach. Firstly, classes in industrial time series are not balanced, because malfunctions are a rare occurrence. Secondly, it is more important to capture real outliers, than to designate the normal value as abnormal. The first problem can be solved with methods such as oversampling, adding artificial anomalies. The second issue should be considered when appropriate metric is being chosen in order to interpret the results of detection correctly [10].

The possibility to use labelled data is not provided often, this is why unsupervised learning methods are in demand. These methods are based on the supposition that normal observations should follow some patterns, which which are not typical for the anomalies. It does not have to be just one pattern. If there are few different patterns, points will be clustered in few groups, but outliers are supposed to be far away from these groups. But there is a huge drawback in such an assumption: normal points could not follow any patterns and be uniformly distributed. As a result, a lot of normal points would be marked as outliers and vice versa. This is why it would be better to use supervised methods with simulated outliers or use approaches which give a possibility to detect outliers without a complete definition of classes, such as some of the proximity-based and clustering-based techniques [10].

Semi-supervised methods are relevant in cases when only a small part of data is labelled. If there are few marked objects, they can be used with unmarked close to them objects for training a model for normal objects. Next, this model can be used to define anomalies, because observations that do not match the model would be marked as anomalies. This approach is also not perfect, since it is not highly possible that not big enough amount of labelled data will represent correctly all probable outliers [10].

Another one approach for the anomaly detection methods classification is based on outliers being compared to the rest of the data. In accordance with this anomaly detection methods can be divided into the following types: statistical, proximity-based, clustering-based.

Statistical methods for anomaly detection assume data normality: normal observations are supposed to be generated by a statistical model and if some objects do not follow this pattern, the are considered as outliers. Statistical methods can be efficient, but it completely depends on how correct the assumption underlying the model is [10].

Statistical approaches are embodied in such prediction-based models as the Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) and their varieties. There is also a Prophet - algorithm created by Facebook for time series forecasting, which can be used for prediction [9]. All these models are pretty simple, fast, can be easily interpreted, but it is not possible for them to capture non-linear dependencies and they are too reliable on the chosen parameters [5]. The key idea of the prediction-base approach is to predict the values for the future period based on the available values and compare the predictions with the actual data in order to decide whether this point is normal or abnormal.

The problem of individual outliers detection can also be solved with proximity-based algorithms. There is a proximity-based definition of point outliers: "The outlier score of a data point is defined in terms of its k-nearest neighbour distance to data points in a time widow of length W"[2]. The performance of proximity-based models depends on the chosen distance measure. Also this methods are not effective enough when there is a a group of anomalous points, in which anomalies are close to each other.

Clustering-based approaches are based on the assumption that all points can be assigned to pretty large clusters, with the exception of outliers, which would be assigned to smaller cluster or would not belong to any cluster. The main idea of this algorithm is to use the distribution difference between normal points and outliers to project them to the vector space. The big disadvantage of using such methods is that they can be computationally expensive and therefore not suitable for too large datasets, it is not possible for them to treat temporal and spatial features of the time sequences.

Nowadays machine learning methods are widely used for solving different tasks in industry, including anomaly detection. Machine learning based algorithms for anomaly detection in multivariate time series could be divided into three categories: clustering-based, prediction-based and classification-based [4].

The traditional machine learning algorithms are supervised usually. There are examples of this approach: Decision Trees, K-Nearest Neighbours, Random Forests, Isolation Forests, etc. The big advantage of these algorithms is that they are fast, efficient and easily interpreted. However, there is also a drawback: for multidimensional data complexity could be too high and performance not as efficient [5].

A possible way to get higher performance is to use deep learning-based approaches, which are becoming more and more effective. There are convolutional neural networks (CNN), recurrent neural networks (RNN) and their hybrids (such as CNN-LSTM, multi-head CNN-RNN) among the popular deep learning methods [5]. The big disadvantage of these algorithms is that it is not possible for them to capture heterogeneity of data sources and temporal correlations. What is more, these approaches assume the presence of huge variety of the data, usually labelled. In the real industrial conditions this becomes a significant obstacle due to the fact, that collecting lots of information about anomalies in order to label the data could be not possible. For example, equipment malfunctions are rare and eventful.

Deep neural networks (DNNs) offer opportunities to capture temporal semantics. DNNs' role in modelling and forecasting time series is growing rapidly. Among the popular models there are Recurrent Neural Networks (RNNs)-based [7], Long Short-Term Memory Networks (LSTMs)-based [8] and hybrids [5].

Among the deep learning architectures, the Autoencoders (AE) are also very effective for anomaly detection in time series. The key idea is that such models consist of two parts: encoder and decoder. The first part performs compression of the input data, the second – decompression to their original shape. The training uses input data that does not contain abnormal observations. The test sample includes data with both normal and abnormal values. The real test data is compared with the reconstructed ones, and the critical difference indicates that they are abnormal.

In addition to this, there are attention-based models improve are widely used with the aim of improving learning ability of complex dependencies. Self-attention-based Transformer can be used for sequence modelling with a high performance due to its ability to understand recurring patterns with long-term addictions [6].

1.3 Metrics for evaluation of the anomaly detection methods in time series

Since the task of determining anomalies in data is often reduced to the task of classification (dividing the data into 2 classes: abnormal ones. normal), then classification metrics are suitable for evaluating the effectiveness of models. Such metrics are widespread, understandable, and convenient for comparing methods.

Classification metrics are calculated based on the confusion matrix values. There are four indicators, forming the confusion matrix: true positives (TP), true negatives (TN), false positives (FP), false negatives (FN). The essence of these indicators:

- TP – the abnormal value was correctly predicted;
- TN – the normal value was correctly predicted;

- FP – the abnormal value was predicted incorrectly;
- FN – the normal value was predicted incorrectly.

The first metric considered in the work based on these indicators is recall. It is calculated as a ratio of the TP to the sum of the TP and FN. The metric demonstrates the sensitivity, true positive rate. In the industrial domain this metric is significantly important, since the main task is to detect all the problems in the industrial process.

The second considered metric is precision, it shows the ratio of TP to the sum of the TP and FP. This metric indicates accuracy in detecting anomalous points.

The third one metric, which combines two previously described is F1-score – harmonic mean of precision and recall. Since the data to solve the problem is unbalanced (many negative cases compared to positive ones) and we need to focus on the positive cases, the F1-score is an appropriate indicator for general model performance estimation.

2 INDUSTRIAL TIME SERIES EXPLORATION

2.1 Data analysis

The specifics of using different methods to determine anomalies in industrial data will be evaluated based on Skoltech Anomaly Benchmark (SKAB) (version 0.9) [11]. Benchmark is one of the largest in the industrial field and include 34 datasets with industrial time series. Collective anomalies and change points are presented in each dataset. There are some important distinguishing feature of this data comparing to other industrial time series:

- the datasets are multidimensional;
- equal time intervals in each time series;
- the data is labelled: anomalies and change points are marked;
- the time series are collected in the process of conducting real experiments

In the field of industry, there are not many datasets that are convenient for evaluating the application of anomaly detection algorithms, since they are mostly not labelled and include not equal time intervals, therefore it is not possible to evaluate performance of the algorithms precisely. Also industrial time series are univariate usually. However, in practice anomalies depend on few indicators more often. For example, the operation of the equipment is usually checked by the values of several sensors responsible for different parameters, since the equipment in industry is a complex system.

What is more, industrial datasets consist of synthetic data often, since there is a lack of real information. The problem is in the difficulties in collecting such a data. Also industrial companies do not share their private information usually. Synthetic data may be suitable for analysing and verifying the anomaly detection methods, but not applicable on real data due to the fact, that synthetic time series may not reflect the real patterns.

For the reasons described above SCAB is well suited for application of anomaly detection and change point detection methods for detecting point outliers and collective anomalies.

Each dataset in the benchmark consists of data collected during experiments conducted in the experiment on the test-bed. The test-bed includes 5 systems: water circulation system, monitoring system, control system, Time-Sensitive Networking technology demonstration system, data system. Experiments were carried out with the help of these systems, the data was collected in a database, and subsequently uploaded to csv files [11].

Each file consists of a data frame with index representing a date and time of writing the value to the database and 10 columns (Table 2.1).

Table 2.1 Columns of the data frames [11]

Name of column	Description
Accelerometer1RMS	Absolute vibration acceleration (Amount of g units)
Accelerometer2RMS	Absolute vibration acceleration (Amount of g units)
Current	Amperage on the electric motor (Ampere)
Pressure	Pressure in the loop after the water pump (Bar)
Temperature	Temperature of the engine body (Degree Celsius)
Thermocouple	Temperature of the fluid in the circulation loop (Degree Celsius)
Voltage	Voltage on the electric motor (Volt)
RateRMS	The circulation flow rate of the fluid inside the loop (Liter per minute)
anomaly	Target, shows if the point anomaly (1) or not (0)
change point	Target, shows if the point is a change point for collective anomalies (1) or not (0)

Total number of the point anomalies in the whole data is equal to 13067, of the change points – 129. The largest data frame consists of 1327 observations, the smallest – of 745 observations.

There is an example of the head of the data frame in the Figure 2.1. You can see that the spread of values for each of the indicators is different. Therefore, it may be necessary to pre-process the data in order for further work with them to be correct.

datetime	Accelerometer1RMS	Accelerometer2RMS	Current	Pressure	Temperature	Thermocouple	Voltage	Volume Flow RateRMS	anomaly	changepoint
2020-03-09 15:56:30	0.027608	0.039203	1.290480	0.054711	68.6194	24.3670	241.062	32.0362	0.0	0.0
2020-03-09 15:56:31	0.027166	0.039940	1.285650	0.382638	68.5923	24.3660	238.709	32.9649	0.0	0.0
2020-03-09 15:56:32	0.027718	0.040167	1.155880	0.054711	68.5207	24.3666	226.485	32.0362	0.0	0.0
2020-03-09 15:56:33	0.028045	0.038026	0.971268	0.382638	68.5425	24.3634	220.378	32.9649	0.0	0.0
2020-03-09 15:56:34	0.027644	0.038580	1.072460	-0.273216	68.6569	24.3639	233.922	32.0000	0.0	0.0

Figure 2.1 – Head of the data frame

As far as we can see, there are not any missing values in the anomaly free dataset datasets. The type of all values is float (Figure 2.2). The analysis showed, that in the other datasets (experiments) the general characteristics remain as follows.

```
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Accelerometer1RMS    9405 non-null   float64
 1   Accelerometer2RMS    9405 non-null   float64
 2   Current              9405 non-null   float64
 3   Pressure             9405 non-null   float64
 4   Temperature          9405 non-null   float64
 5   Thermocouple         9405 non-null   float64
 6   Voltage              9405 non-null   float64
 7   Volume Flow RateRMS  9405 non-null   float64
dtypes: float64(8)
```

Figure 2.2 – Information about anomaly free dataset

Data visualization of the random dataset is presented in the Figure 2.3. On each graph, the values of the indicator are postponed along the X axis, and the corresponding time is on the y axis.

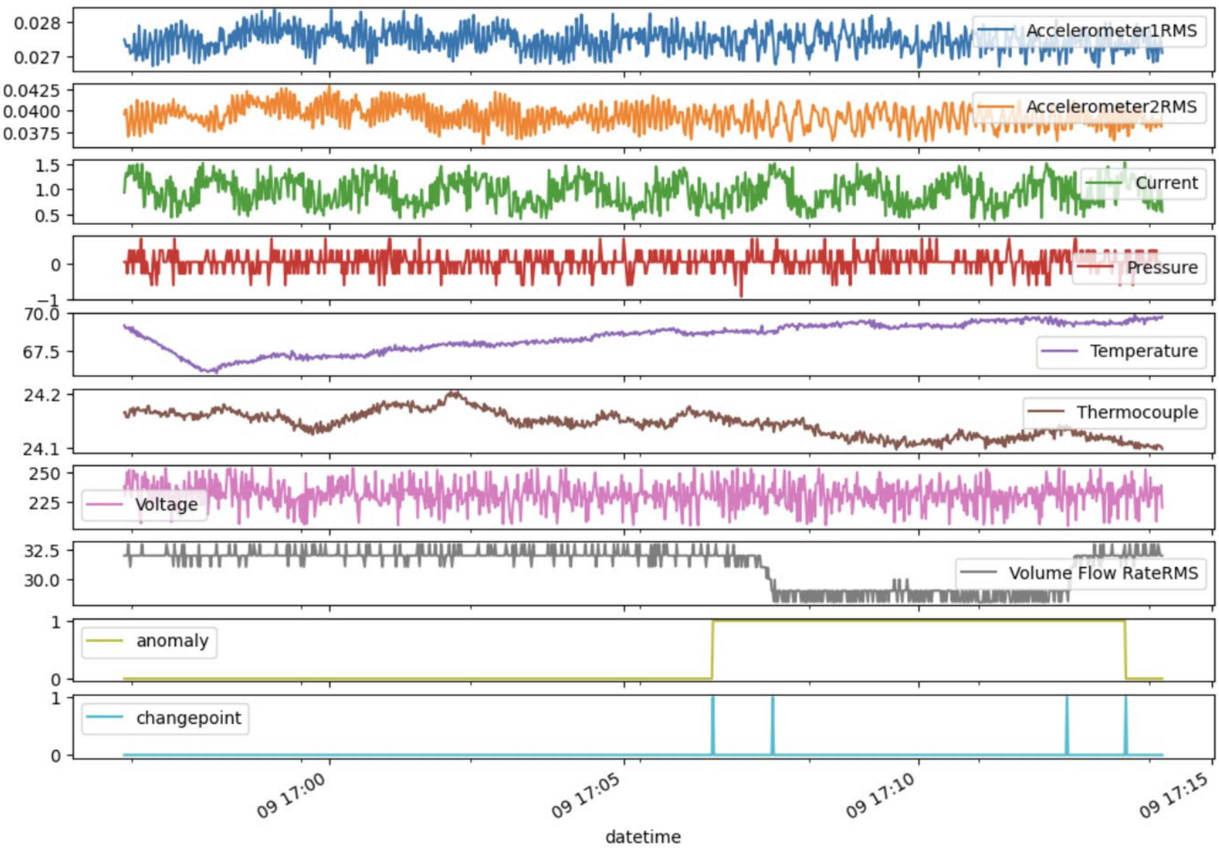


Figure 2.3 - Visualization of the dataset

According to the graphs presented above, it can be seen that some indicators seem to be quite stable, within certain limits, for example: accelerometer1RMS, accelerometer2RMS, current, pressure, voltage. Other indicators, such as temperature, thermocouple and volume flow rateRMS, are undergoing changes. We can see, that these changes could be connected with the behaviour of the target variables anomaly and change point. When temperature grows more rapidly, thermocouple begins to decrease, at the same time volume flow rateRMS falls it falls and remains at an extremely low level. During these changes the appearance of anomalies is observed.

The change point value is helpful for indicating different stages of the system work. There is a healthy stage, when system is stable, in this stage the change point's value equals to 0. After that values of some of the indicators begin to deviate from the normal, the change point equals to 1 and transition stage begins. After transition stage finishing the anomalous stage begins, another one change point appears. The duration of anomalous stage is about 5 minutes. After this stage the third change point appears, indicators return to the normal behaviour, transition state begins. Then, another one change point appears and the process returns to the normal stage [11].

In order to analyse the pairwise relations we can visualise a matrix of scatter pair plots in the random dataset (Annex A). By estimating the distribution of each variable, we can conclude that the distribution of only one variable (accelerometer1RMS) is close to normal. In the some of the scattering diagrams of the relationships between each paired combination of variables we can see outliers, which are far away from the vast majority, for example in the relation of thermocouple and current.

We will estimate the correlation between the variables in the chosen dataset using a thermal diagram (heat map), presented in the Annex B. There is a correlation higher than 50% between the temperature and target variable, thermocouple and volume flow rateRMS. The connection between temperature and thermocouple, temperature and volume flow rateRMS, thermocouple and target variable, volume flow rateRMS is characterized by a negative correlation. These results confirm the behaviour of the indicators on the chart (Figure 2.3).

It is important to notice, that the correlation of variables is very different in different datasets according to the data analysis. This may affect the effectiveness of using the same anomaly detection methods when conducting different experiments.

In accordance with the behaviour of the variables shown in the Figure 2.3, we can assume, that some of the time series are not stationary. In order to check this fact we well perform the extended Dickey-Fuller test, which uses the following hypotheses:

- H_0: the series has a certain time-dependent structure and does not have constant variance over time;
- H_1: the time series is stationary.

If the p-value from the test is less than a certain level of significance (for example, $\alpha = 0.05$), then we can reject the null hypothesis and conclude that the time series is stationary.

The results of the test showed that the following time series are not stationary in the all datasets: temperature, thermocouple, volume flow rateRMS. But it is important to remember that in addition to the test results, a unique problem and expertise in the subject area should be taken into account. A non-stationary time series may give the impression that there is a high correlation between two variables, which we could examine in the Figure 2.3 and correlation matrix (Annex B), but in fact the correlation exists only because of a trend, seasonal element in the data or some unjustified changes in the process.

There is a variety of methods for making time series stationary, such as differentiation, Seasonal Time Series Decomposition (STL), logarithmic transformations. However, maintaining a real stationarity of time series can be a better approach, since the detected pattern could describe some meaningful information. For this reason, the stationarity of the time series will not be changed in the analysed data.

2.2 Data preprocessing

The ratio of abnormal and normal values in the all data frames is shown in the Figure 2.4. We can conclude, that the proportion of anomalies ranges approximately from 25% to 45% in different experiments. From the point of view of the classification problem, datasets are unbalanced.

In accordance with Figure 2.4, the proportion of anomalies in some datasets is significantly higher than in most. Datasets in which the proportion of anomalies is no more than 40% will be left for further work, since most of the datasets are imbalanced.

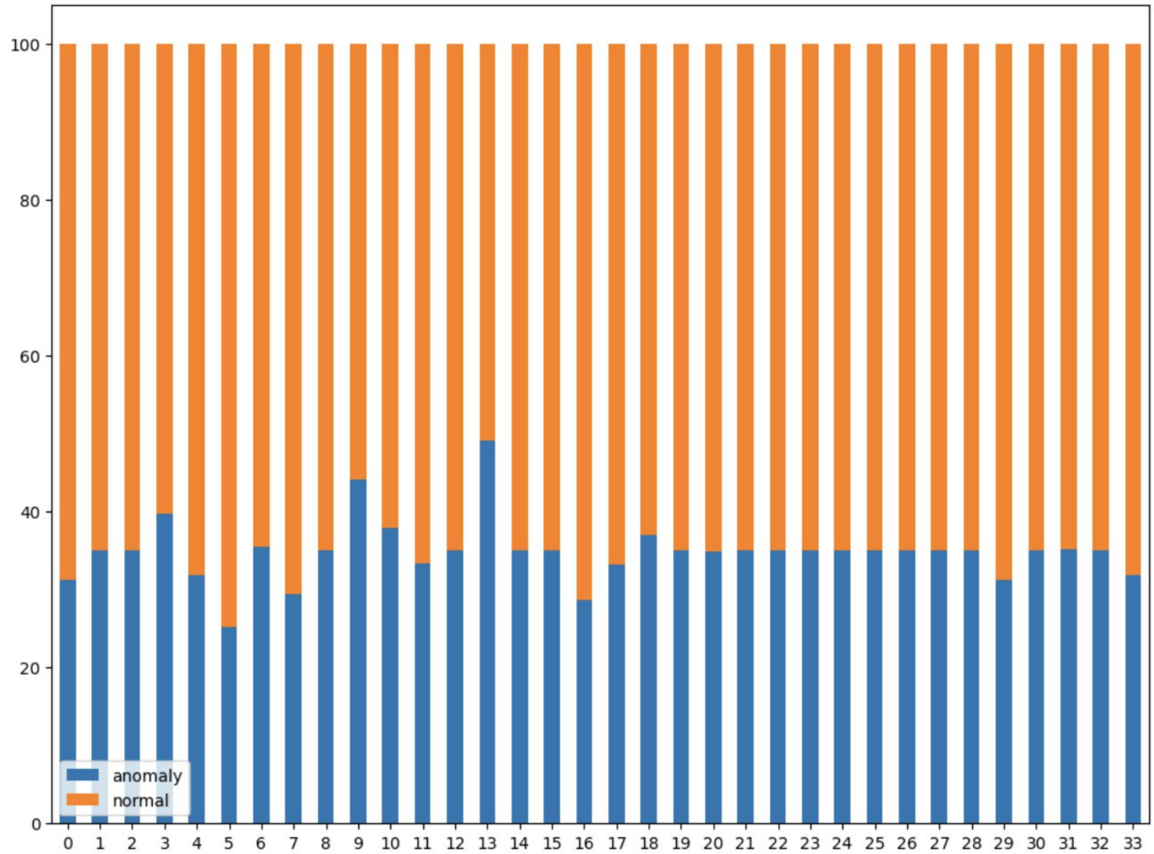


Figure 2.4 - The ratio of abnormal and normal values in the datasets

Models will be trained on the data without anomalies, due to the fact, that some models (like reconstruction-based) should see only normal data while training. The analyses of all the data frames has shown, that the earliest anomaly occurs in the 495th observation. Then, we will use first 480 observations of each dataset for training and the rest of the data for predicting and testing.

As it was noticed earlier, different indicators in the datasets take values from very different limits in the datasets. This is why we will use StandardScaler before detecting anomalies with some of the algorithms. This method as a preprocessing tool, will allow us to standardize the range of functionality of the input set. The key idea of the method is to change the size of the distribution of values so that the average value of the observed value is 0 and the standard deviation is 1.

3 APPLICATION OF MACHINE LEARNING APPROACHES FOR ANOMALY DETECTION IN INDUSTRIAL TIME SERIES

3.1 Isolation Forest

Industrial data describes complex processes and systems. As a result, most of the real world industrial time series are multi-dimensional and require unsupervised anomaly detection techniques. This is why machine learning approaches are widely used for anomaly detection in industrial time series. Such methods, unlike statistical ones, allow us to determine outliers and anomalies effectively not only in one-dimensional data.

We will start exploring machine learning models with unsupervised classification algorithm – Isolation Forest, which is widely used in problems of detecting anomalies in various types of data. This algorithm was developed by Fei Tony Liu in 2008 [12], it remains in demand at the present time.

Isolation forest is an example of ensemble model, which combines the results obtained from numerous decision trees. Algorithm is fast, as it splits the data randomly, works with multi-dimensional data, requires low memory. The key idea of the algorithm is that anomalous points can be more easily separated from the others.

Anomaly detection with algorithm includes two steps:

1) building Isolation Trees using a training data;
2) assigning an anomaly score to each instance of the testing data passed through built Trees.

First of all, we will define the model and analyse its performance on the dataset used for an example during data analysis (fourth data frame in the whole data). Model is defined with random state (equals to 0), since time series include temporal dependencies and the order is important. The contamination parameter controls the decision function threshold. Since, the task of anomaly detection is supposed to be unsupervised in the industrial domain, the amount of point anomalies in the data should be unknown. This is why contamination hyper parameter has been optimized with the RandomizedSearchCV – the method for hyper parameters optimization implemented in the scikit-learn library. As the result, the value of the contamination parameter was set to 0.01 with accordance to the best f1-score achieved.

The results of Isolation Forest anomaly detection obtained after training and model prediction are shown in Figure 3.1

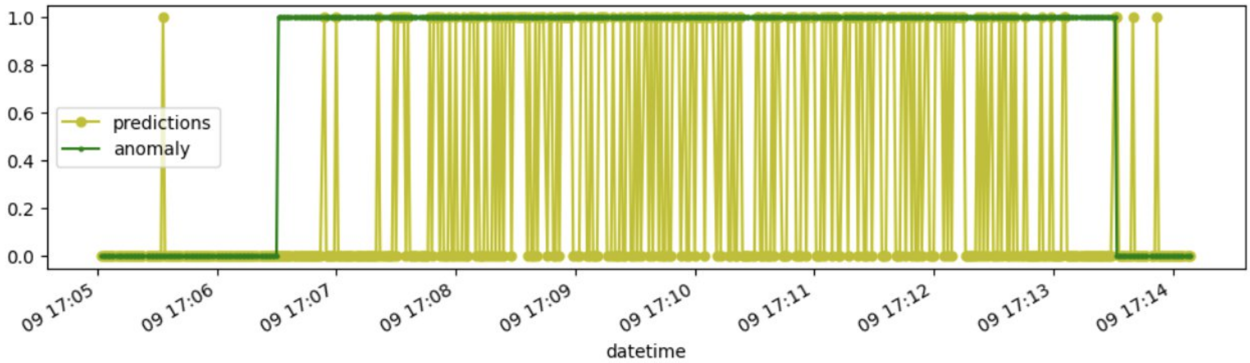


Figure 3.1 – Predicted anomalies (Isolation Forest)

Based on the results shown in Figure 3.1, it can be seen that the algorithm was able to detect a large number of abnormal points. An interval consisting of the vast majority of predicted anomalous values is narrower than an interval including real anomalies. Moreover, some points are detected as anomalies, but in fact they represent normal values and do not belong to the abnormal stage (Figure 3.2).

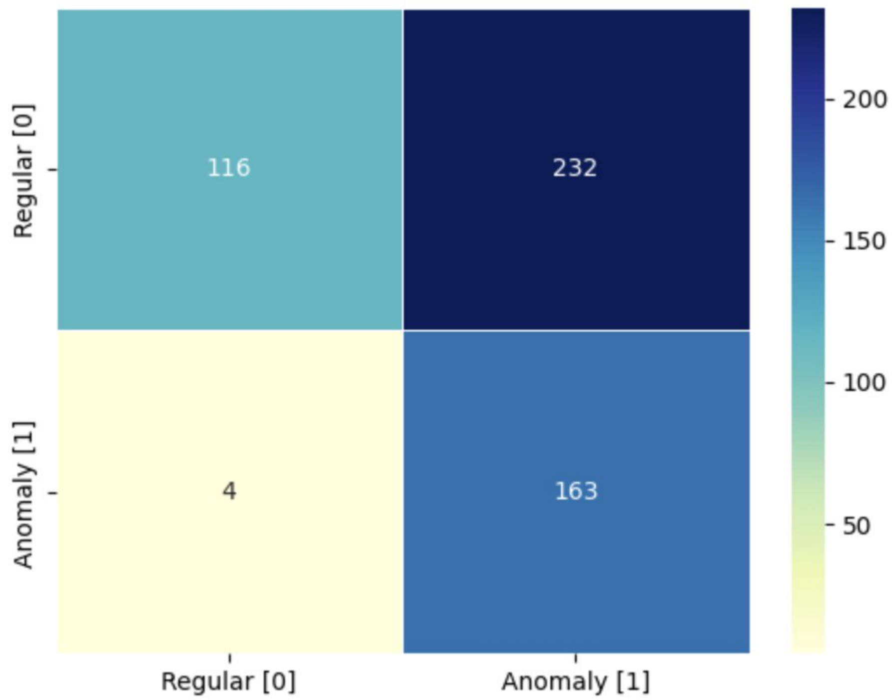


Figure 3.2 – Confusion matrix (Isolation Forest)

There are 116 true negatives, 4 false positives, 232 false negatives and 163 true positives detected by the model according to the Figure 3.2. We can see, that model made a lot of mistakes, but it is important to mention, that algorithm has detected the biggest amount of anomalies during the abnormal stage. As a result, it could be helpful for detecting the fact, that problems have been occurred during systems work.

Detecting a moment, when process stopped being stable, is even more crucial issue than detecting point anomalies sometimes, especially in the industrial domain. The predicted change points are presented in the Figure 3.3.

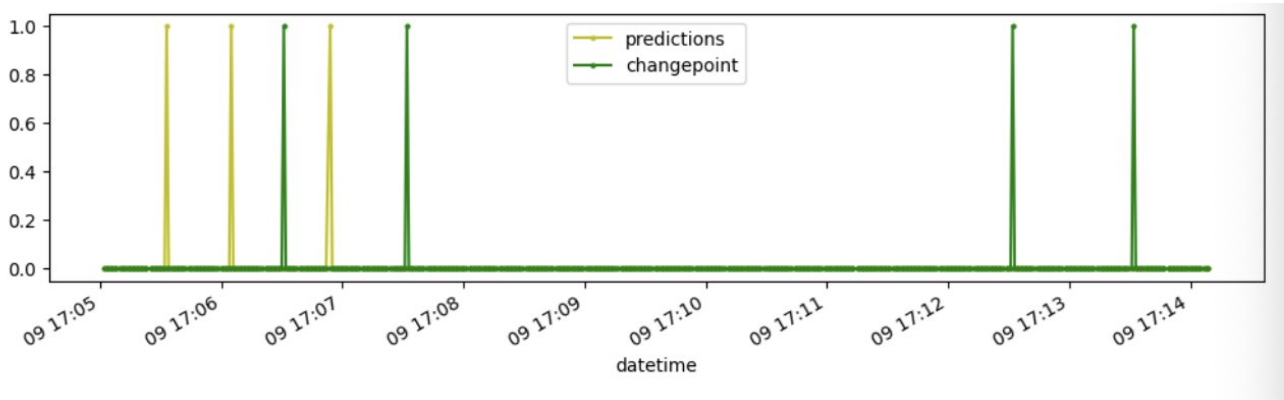


Figure 3.3 – Predicted change points (Isolation Forest)

Again, the model was not accurate enough in determining the change points. It detected change points much more earlier than they occurred in real data. Detection of the first change point is the most important, since it is a signal, that anomalous stage has begun.

We can estimate general performance of the model on the examined data frame with classification report (Figure 3.4). The precision is higher and close to 100% for the normal data and shows good ability to distinguish regular data from the with abnormal. The recall is also very high for anomalous observations, showing ability for detecting anomalies. F1-score is a little bit higher than 50%, which confirms the low quality of anomaly detection by the model.

	precision	recall	f1-score	support
0	0.97	0.33	0.50	348
1	0.41	0.98	0.58	167
accuracy			0.54	515
macro avg	0.69	0.65	0.54	515
weighted avg	0.79	0.54	0.52	515

Figure 3.4 – Classification report (Isolation Forest)

The results obtained using the example of a random dataset may not reflect the actual effectiveness of the algorithm. We will determine the values of the metrics for the data collected during all the experiments (Table 3.1).

Table 3.1 - Metrics for all datasets (Isolation Forest)

	F1-score	Precision	Recall
Average value	0.51	0.62	0.60
Minimum value	0.28	0.30	0.35
Maximum value	0.83	0.90	0.82

It can be concluded that the indicators are not stable, since the minimum and maximum values are very different. This may indicate that the algorithm is more suitable for some experiments than for others. Such results are due to differences in the ratio of indicators in different experiments, which was noticed when analysing the data. At the same time, the average value of F1-score is close to one obtained for random data frame.

In the Annex C we can find the visualisations of the data features from the dataset, which shows the best results (Figure C.1) and the dataset, which shows the worst results (Figure C.2). It can be noted that in the dataset that demonstrates the best results, during the abnormal stage, the behaviour of indicators such as accelerometer1RMS, accelerometer2RMS and volume flow rateRMS varies greatly in comparison with the normal stage unlike other indicators, the behaviour of which is more stable throughout the period.

At the same time, the indicators of the dataset with the worst results do not reflect the changes associated with the transition to the abnormal stage at all. Consequently, the poor performance of the model may be due to inefficient operation of sensors or poor-quality marking, which is really very labour-intensive in industry.

3.2 One-Class Support Vector Machine

In this section we will estimate the performance of another one model appropriate for anomaly detection – One-Class Support Vector Machine [13]. The important characteristic of this model is that it is aimed at eliminating the class imbalance focusing on the majority class, especially in anomaly detection.

Traditional Support Vector Machines are used for classification, but OCSVM is effective in detecting patterns not similar to normal in unsupervised manner. The algorithm is relevant for cases when the training sample contains only normal data, and the test sample contains abnormal data too. It means, that for implementing this algorithm we should be confident, that data is normal. We use labelled datasets, this is why this requirement will be fulfilled.

The main idea of the algorithm is to divide classes by a hyperplane so as to maximize the distance between them. One of the model parameters is kernel, which enables the model to establish non-linear dividing boundaries.

Optimization of the hyperparameters with the GridSearchCV and RandomizedSearchCVdid not significantly improve the result in comparison with the baseline model.

We will start with applying OCSVM to the indicative dataset, which was used for data analyses earlier. The result of point anomalies prediction is presented in the Figure 3.5. We can see, that results are less accurate than those one obtained with Isolation Forest (Figure 3.1). Model was not effective enough in detecting change points, but it found 2 of 4 change points before they occurred (Figure 3.6).

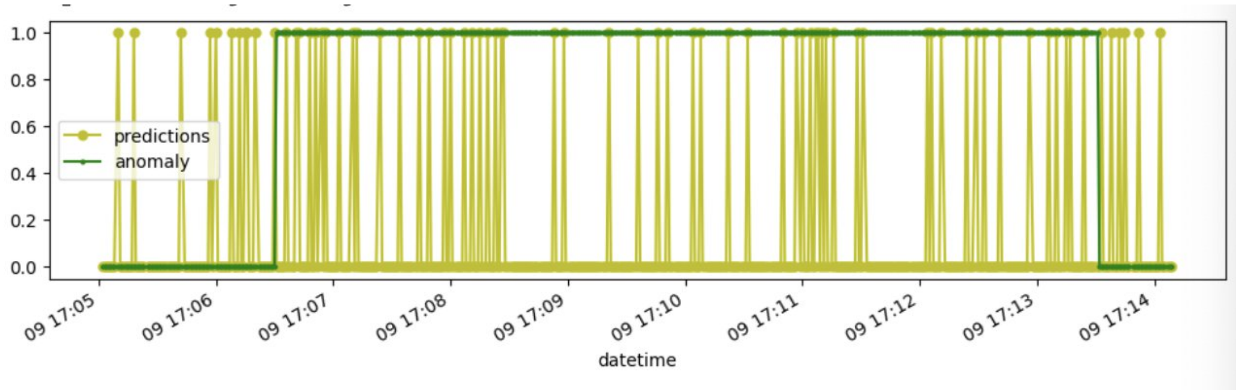


Figure 3.5 – Predicted anomalies (OCSVM)

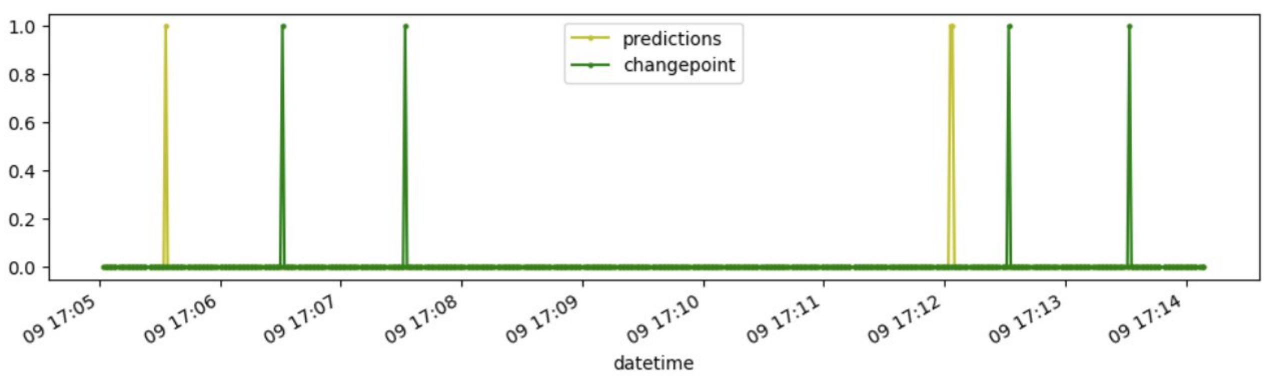


Figure 3.6 – Predicted change points (OCSVM)

It is not possible to detect anomalous stage, since there are some groups of predicted anomalies close to each other, the anomalous points are densely distributed throughout the segment.

For estimated metrics the following results were obtained: f1-score - 0.31, precision score - 0.50, recall score - 0.50. It can be seen that the values were significantly lower than values obtained for Isolation Forest, general performance of OCSVM is poor for analysed data.

The metrics for whole data are presented in the Table 3.2. They do not differ much in comparison with the results of the dataset from the example. The recall demonstrates the highest level. The amount of true positives equals to 57, false positives - 17, true negatives - 103, false negatives - 338.

Table 3.2 - Metrics for all datasets (OCSVM)

	F1-score	Precision	Recall
Average value	0.39	0.43	0.50
Minimum value	0.31	0.27	0.47
Maximum value	0.43	0.55	0.53

3.3. K-Nearest Neighbour

The K-Nearest Neighbour (KNN) is a machine learning algorithm which is mainly used for classification and regression tasks, it is supervised approach usually. This method implements classification process according to the neighbours of each point. However, this method can be used for anomaly detection tasks.

We will use KNN model implemented by PyOD library [14], which is appropriate for unsupervised tasks. This approach works similarly to clustering task: belonging of each point to the normal or abnormal group depends on its distances to k-nearest neighbours, if distance is large than assumed to be, the point will be classified as anomalous. The hyper parameters were chosen with the RandomizedSearchCV method in accordance with the best f1-score achieved. Contamination hyper parameter was set to 0.1.

Point anomalies, detected by KNN models are presented in the Figure 3.7. There are some points, which are not anomalies, but were detected as abnormal. However, we can see the detected anomalous stage in the chart. The determined stage is shorter than real, but the biggest part of points belonging to this stage is detected correctly.

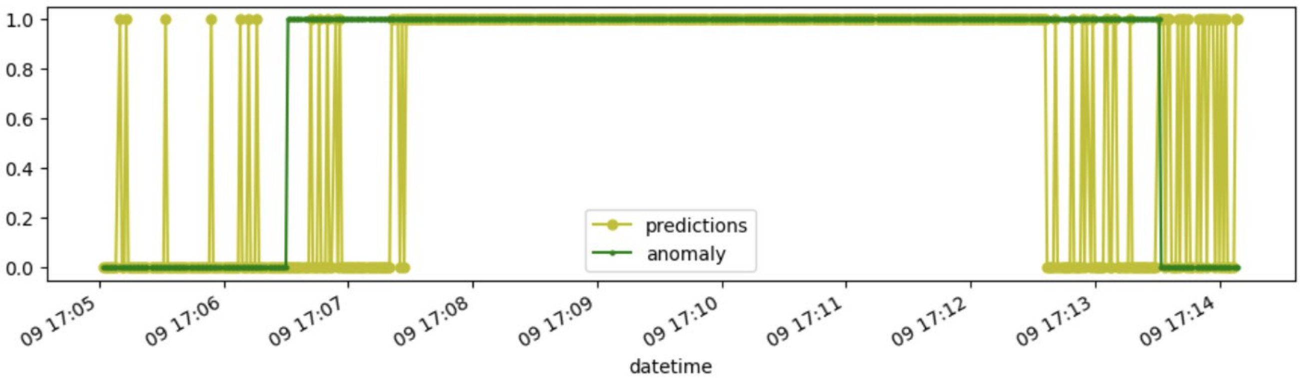


Figure 3.7 - Predicted anomalies (KNN)

In accordance with confusion matrix, presented in the Figure 3.8, there are 94 true negatives, 26 false positives, 84 false negatives and 311 true positives. We can assume, that model made mistakes less often, than correct predictions.

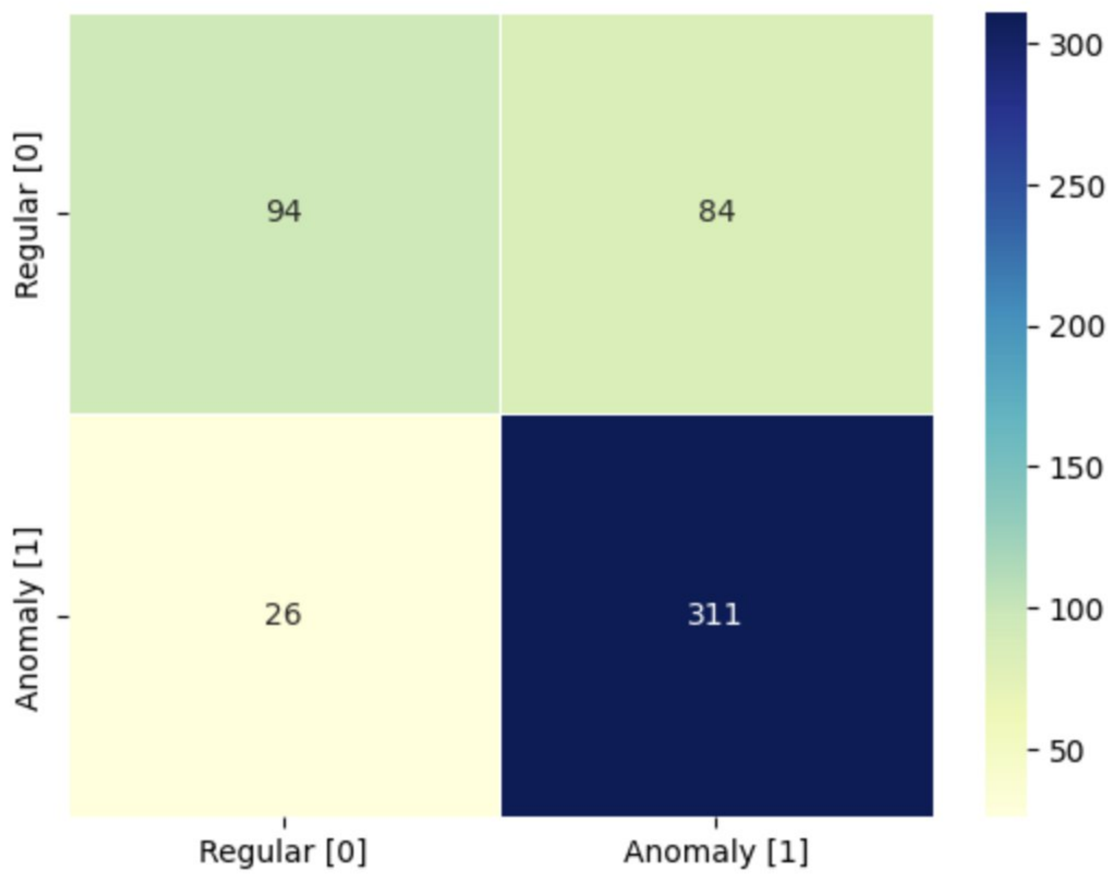


Figure 3.8 – Confusion matrix (KNN)

For the KNN model applied to the examined dataset the following metrics were obtained: F1-score – 0.74, precision score – 0.73, recall score – 0.79. It can be concluded, that this result is the best among all previously examined models. What is more, we see, that metrics are pretty balanced. The results obtained with the KNN model seem to be more reliable.

We can find predicted change points in the Figure 3.9. Algorithm has detected only one change point of 4 and much more earlier than it occurred.

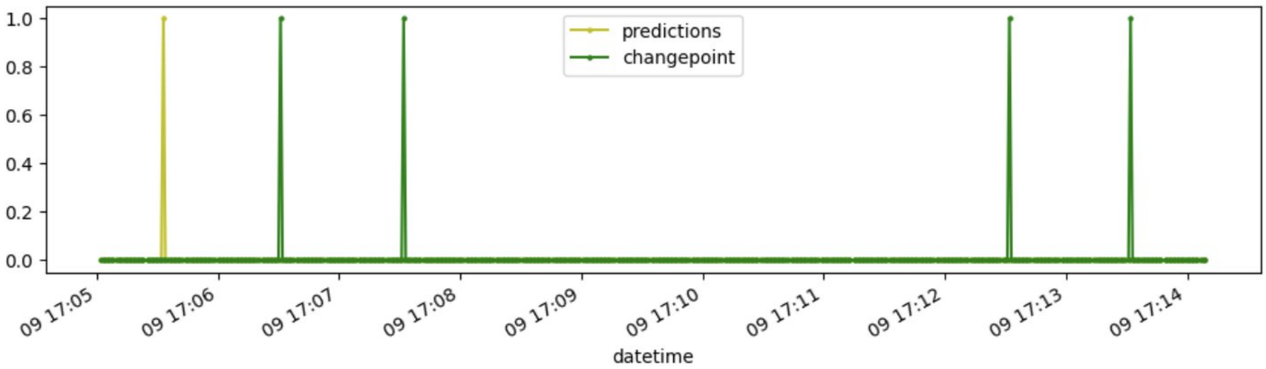


Figure 3.9 – Predicted change points (KNN)

Analysing the metrics obtained after applying the model to all the data, we can see that F1-score (Table 3.3) is lower, than value obtained in dataset used for example. Once again, the reason for such result can be connected with peculiar properties of the features indicated during different experiments.

Table 3.3 - Metrics for all datasets (KNN)

	F1-score	Precision	Recall
Average value	0.62	0.71	0.69
Minimum value	0.29	0.21	0.40
Maximum value	0.91	0.92	0.92

It is noteworthy, that datasets, on which KNN model has shown the best and the worst performance (Annex D) are not the same with the best and the worst datasets with accordance to Isolation Forest model (Annex C). This confirms the idea that the data has different interconnections and characteristics, therefore, for each dataset, the most effective model may differ.

3.4. Convolutional Autoencoder

As was mentioned in the theoretical part of the current work, deep learning methods for anomaly detection in time series are mainly divided into prediction-based and reconstruction-based. Predictive models (CNN, RNN, LSTM) are effective for the considered problem, but they can be limited by the complexity of the production process and not easily trained because of rareness of anomalies in industrial data [15]. The Autoencoders (AE) can overcome the described difficulties.

Autoencoders include two networks connected in series. The first one is an encoder, which compresses the input data in the space of smaller dimension. The second one is decoder, which tries to restore compressed input data. An important feature of AEs is that they do not require abnormal data for training, they can be trained with only regular observations. This characteristic is very important in the industrial domain. It is expected, that time series will not be reconstructed correctly, since time series consist of anomalies. As a result, the loss function (Mean Square Error) will be too high (3) [15].

$$Loss\ (X, \widehat{X})\ <\ Loss\ (A, \widehat{A}), \forall x \in X, \hat{x} \in \widehat{X}, a \in A, \hat{a} \in \widehat{A}, \quad (3)$$

where X – a dataset of normal samples, $X \in R^{t \times m}$ with time steps $t \in N$, and the measurements $m \in N$;

$\widehat{X}$ – reconstructed normal sample after decoding;

A – dataset of normal and anomalous samples;

$\widehat{A}$ – reconstructed sample with anomalies.

The observations in testing data are marked as normal or abnormal depending on the threshold (4) [15].

$$c(A)\ =\ \begin{cases} Normal, if\ Loss\ (A, \widehat{A})\ <\ e_{th}, \\ Abnormal, if\ Loss\ (A, \widehat{A})\ \geq\ e_{th} \end{cases}, \quad (4)$$

where $c(A)$ – an assigned class (normal or abnormal) to observation,

e_{th} – chosen threshold.

Convolutional Neural Networks enable to perform unsupervised feature extraction from the input data. In the current work CNN AE include convolutional layers with, which are used for extracting features and present themselves a part of the encoding stage. Transpose convolutions are used for up-sampling in the decoding stage. Sequential model consists of layers with a kernel size equal to 7, strides equal to 2, filters equal to 32 and 16. Activation function is ReLU. The reconstruction loss is defined with accordance to the mean absolute error loss function.

The detected anomalies in the analysed data after application the CNN AE are presented in the Figure 3.10. Not all point anomalies were determined, but an anomalous stage was detected, its duration is less than real duration. Also there are not normal points which were detected as anomalous at all. It seems that model could be effective for the anomaly detection tasks for which it is important not do not skip normal values.

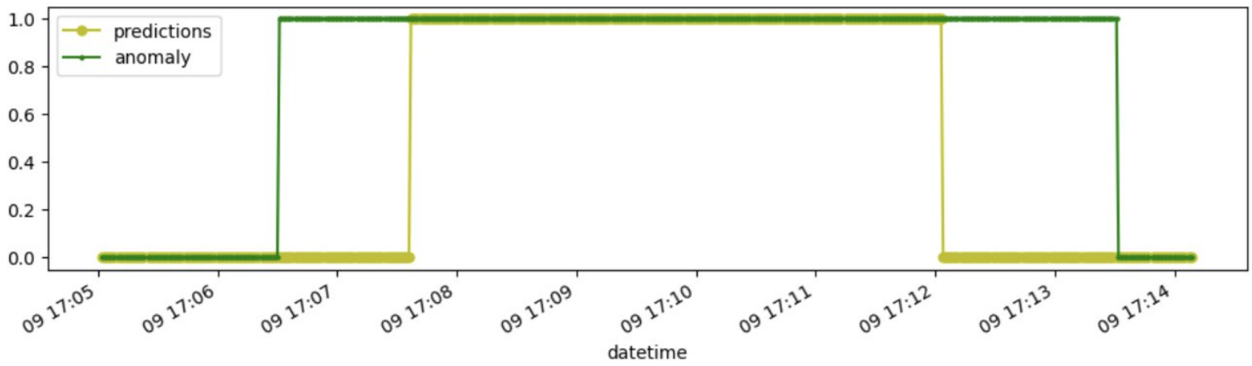


Figure 3.10 – Predicted anomalies (CNN AE)

It can be noted, that 2 of 4 change points were detected. The first one is pretty close to the real change point, but determined later than real. The second one not as close and determined much more earlier than real.

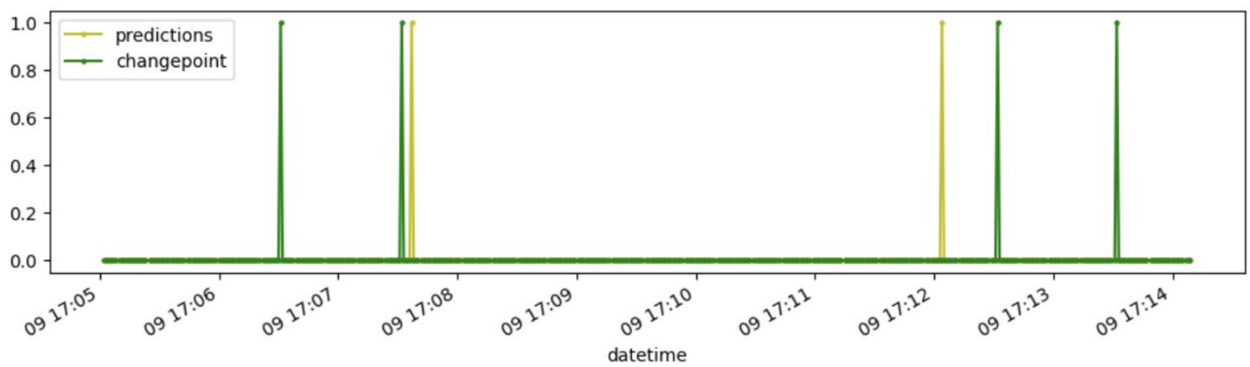


Figure 3.11 – Predicted change points (CNN AE)

The fact that the model makes mistakes less often than the previously studied algorithms (Isolation Forest, OCSVM, KNN) is confirmed by the confusion matrix (Figure 3.12). There are 120 true negatives, 0 false positives, 144 false negatives, 251 true positives. The amount of correct predictions is higher than the amount of false predictions.

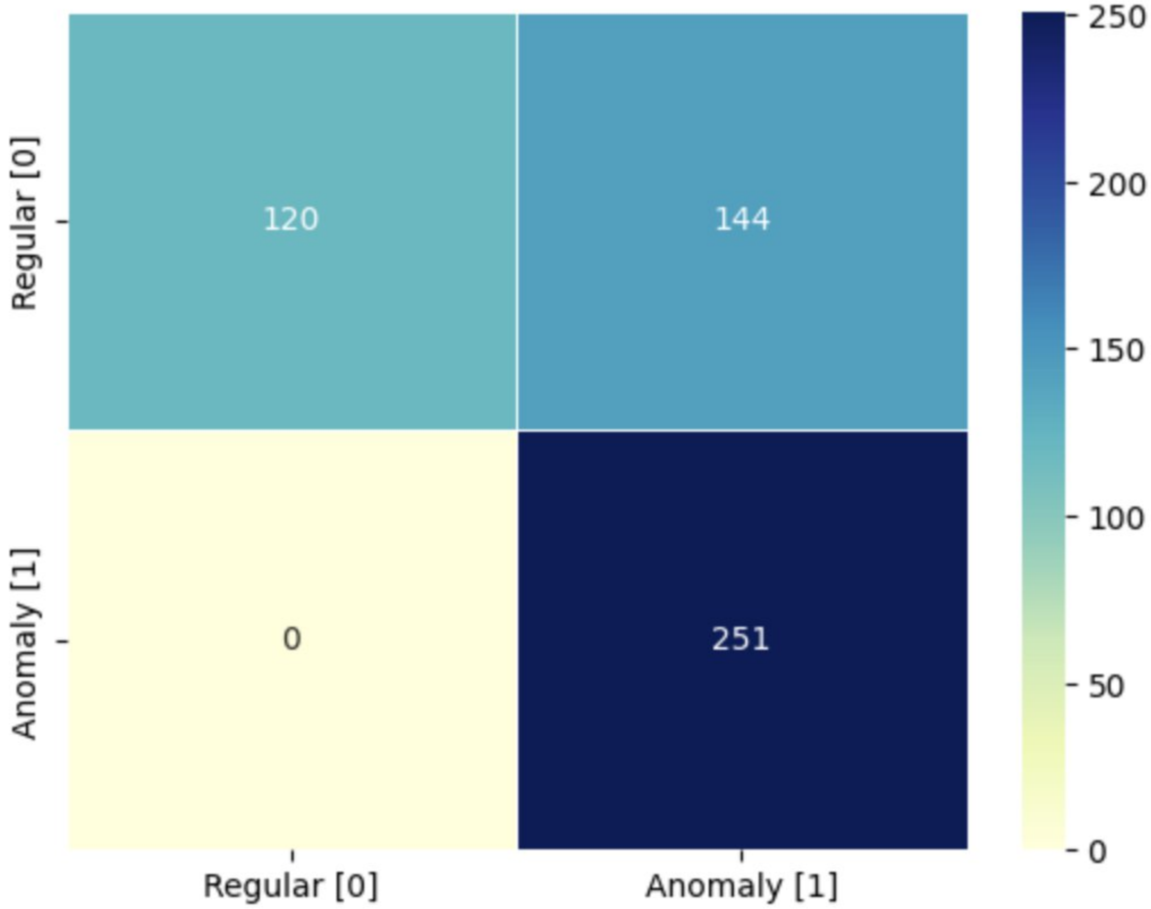


Figure 3.12 – Confusion matrix (CNN AE)

We can conclude, that metrics for the examined dataset also indicate better results (Figure 3.13). The general F1-score equals to 0.70, precision score – 0.73, recall score – 0.82. For the normal data precision is 100%, recall – 45%, for the anomalous data precision is 64%, recall – 100%. As we can see, average metrics are balanced. The highest score is defined for recall, which shows that model defined a vast majority of anomalies.

	precision	recall	f1-score	support
0	1.00	0.45	0.62	264
1	0.64	1.00	0.78	251
accuracy			0.72	515
macro avg	0.82	0.73	0.70	515
weighted avg	0.82	0.72	0.70	515

Figure 3.13 – Classification report (CNN AE)

The results obtained after testing CNN AE on the all data frames are presented in the Table 3.4. We can see, that average values for all classification metrics are close to 70%. This result is the most stable, F1-score is the highest in comparison with algorithms implemented in the previous sections.

Table 3.4 - Metrics for all datasets (CNN AE)

	F1-score	Precision	Recall
Average value	0.69	0.72	0.75
Minimum value	0.28	0.19	0.50
Maximum value	0.93	0.95	0.93

The minimum and maximum results for the former vary greatly, which is explained by the variety of data. The best result (more than 90% for all metrics) indicates the excellent performance of the model.

The best result was achieved for the dataset, which is visualised in the Figure E.1 (Annex E). Few variables in this data (temperature, thermocouple and volume flow rateRMS) change behaviour rapidly, when anomalous stage begins. Data visualised in the Figure E.2 (Annex E) demonstrate drastic changes for features temperature and thermocouple, but this changes happen during the normal stage, what can prevent the model from detecting anomalies correctly, as a result, performance of model for this data is the worst.

3.5 Long Short-Term Memory Recurrent Neural Networks

Autoencoders based on the Long Short-Term Memory Recurrent Neural Networks (LSTM) consist of encoder and decoder. This architecture is powerful for detecting anomalies even if the variables in time series change very frequently.

The LSTM autoencoder learns to reconstruct input sequences. The important characteristic of the LSTM layers – their possibility to capture temporal dependencies in the data. The main advantage of the LSTM comparing to classical RNNs is that it uses “gates” for deciding what information is important and may be allowed for updating the cell state. In the Figure 3.14 an example of the LSTM unit is presented (c_t – cell, i_t – input gate, o_t – output gate, f_t – forget gate) [15].

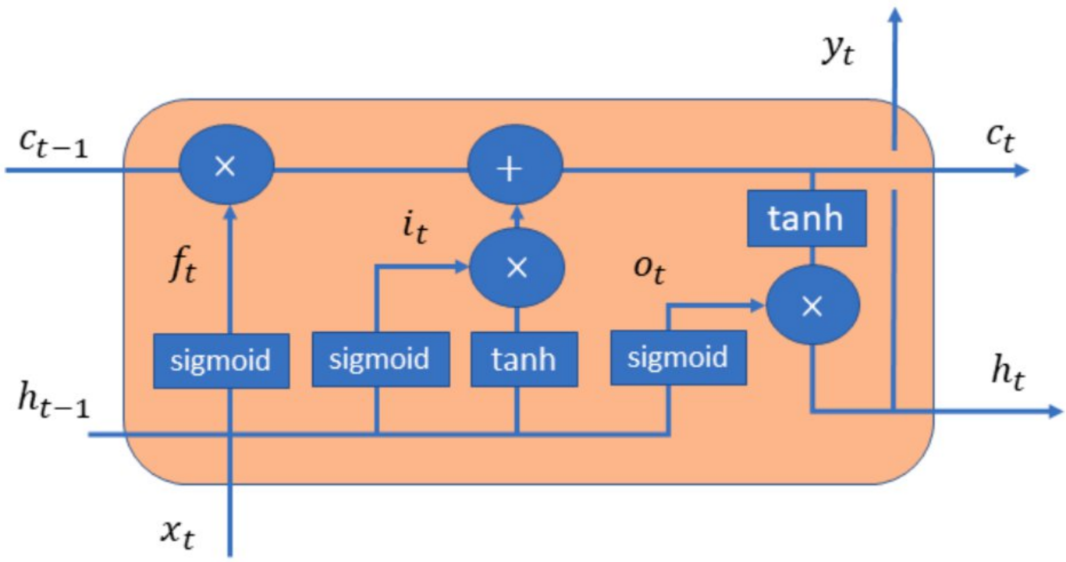


Figure 3.14 – LSTM unit [15]

The LSTM AE model used in this work includes 2 LSTM layers for performing encoding and decoding process. It also includes a layer for repeating the vector in the appropriate timestamps. Activation function is ReLU, loss function is mean absolute error, optimizer - “adam”.

Figure 3.15 illustrates point anomalies, predicted by LSTM AE. We can mention, that result is visually quite similar to those one obtained with CNN AE (Figure 3.10). The anomalous stage is detected, but interval is not appropriate. There are some normal points which were detected as anomalous after abnormal stage was finished.

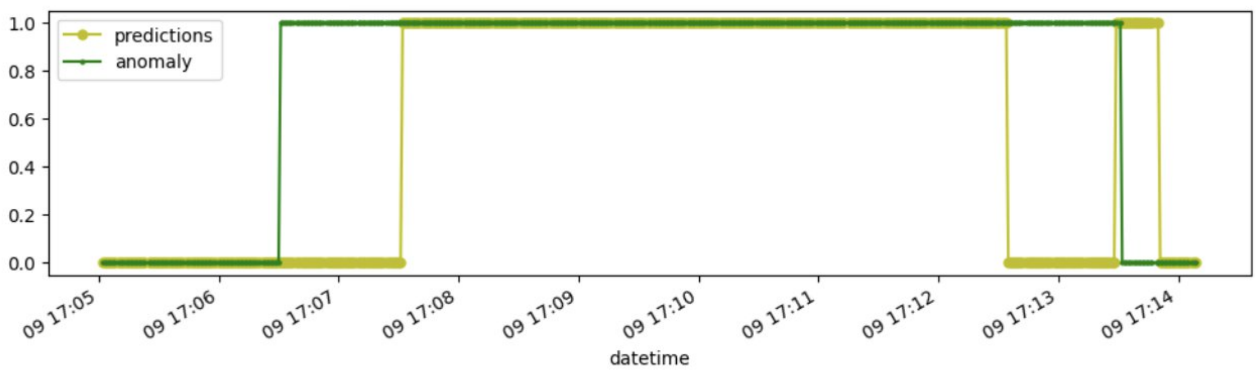


Figure 3.15 – Predicted anomalies (LSTM AE)

There are 3 of 4 detected change points (Figure 3.16), 2 of them are very close to correct time, which is good result. The third change point is not correct at all.

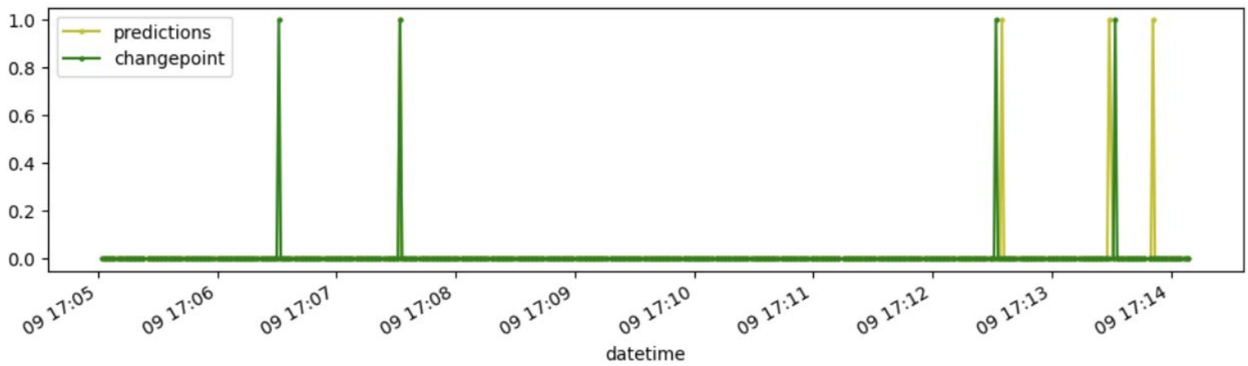


Figure 3.16 – Predicted change points (LSTM AE)

The model makes mistakes less often than makes correct predictions. The confusion matrix demonstrates, that amount of true negatives is 102, false positives – 18, false negatives – 108, true positives – 287 (Figure 3.17). There are some points which were abnormal, but were indicated as regular. In the industrial domain this problem could be crucial, since it is essential to detect hardware problems as early as possible.

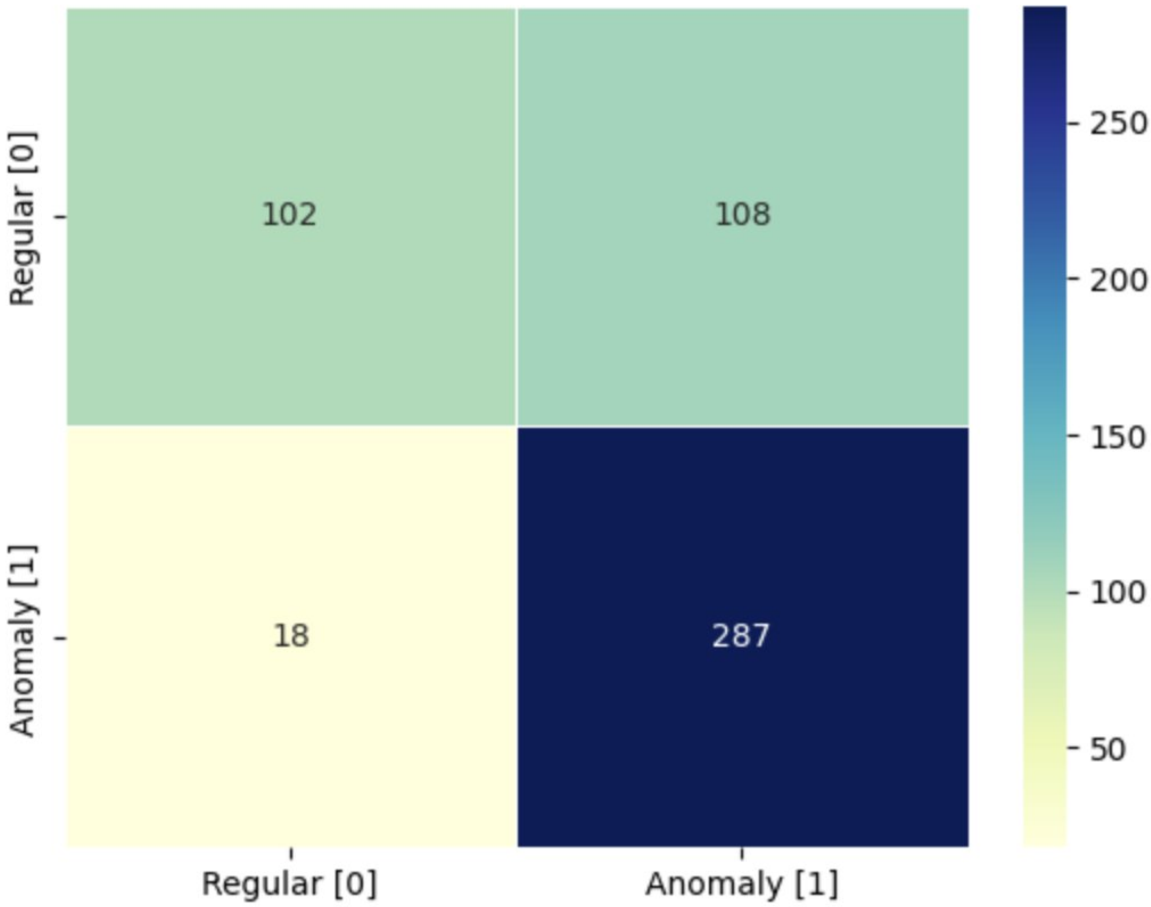


Figure 3.17 – Confusion matrix (LSTM AE)

The classification report for the analysed dataset is presented in the Figure 3.18. The effectiveness of the model in detecting anomalies is higher than in detecting normal points. The general F1-score for this experiment equals to 0.72, precision score – 0.71, recall score – 0.79. Result are balanced, close to the obtained with CNN AE and higher than results obtained with machine learning approaches (Isolation Forest, KNN, OCSVM).

The LSTM AE was also estimated on the whole data (Table 3.5). It can be occluded, that all classification metrics remain on the level close to 65%.

```
              precision    recall  f1-score   support

           0       0.85      0.49      0.62       210
           1       0.73      0.94      0.82       305

    accuracy                           0.76       515
   macro avg       0.79      0.71      0.72       515
weighted avg       0.78      0.76      0.74       515
```

Figure 3.18 – Classification report (LSTM AE)

Table 3.5 – Metrics for all datasets (LSTM AE)

	F1-score	Precision	Recall
Average value	0.64	0.68	0.68
Minimum value	0.22	0.19	0.28
Maximum value	0.92	0.93	0.92

The obtained average metrics for LSTM AE are lower than average metrics for CNN AE. The results demonstrate very different values. The dataset, for which the best performance was achieved, includes stable patterns of the vast majority of features, only one variable changes rapidly, when anomalous stage begins – thermocouple (Figure F.1, Annex F). The worst performance was demonstrated for the dataset, visualised in the Figure F.2 (Annex F). That was exactly the same dataset, which showed the worst result with Isolation Forest (Annex C).

CONCLUSION

Various approaches to the anomaly detection in time series were considered in this work. Generally, approaches can be divided into statistical and machine learning The analysis carried out in the work focuses on the application of machine learning methods in industrial multidimensional time series.

The analysis of the data showed that the behaviour of the variables obtained in individual experiments can be very different. In some experiments, the occurrence of a malfunction in the water circulation system is affected by a change of the Volume Flow Rate indicator, in others – by a temperature, thermocouple or accelerometers. Consequently, the same model can show different results for different experiments, and in one case be effective, and in the other – unable to detect anomalies correctly.

5 models were compared for all the experiments. The average results obtained for all data and individual results for dataset №4, which was chosen randomly for more detailed analysis, are presented in the Table below.

Table – Final results

Model	F1-score		Precision		Recall	
	Average (all datasets)	Dataset № 4	Average (all datasets)	Dataset № 4	Average (all datasets)	Dataset № 4
Isolation Forest	0.51	0.54	0.62	0.65	0.60	0.70
OCSVM	0.39	0.31	0.43	0.50	0.50	0.50
KNN	0.62	**0.74**	0.71	**0.73**	0.69	0.79
CNN AE	**0.69**	0.70	**0.72**	**0.73**	**0.75**	**0.82**
LSTM AE	0.64	0.72	0.68	0.71	0.68	0.79

F1-score, precision and recall were calculated for the models performance evaluation. As we can see, the best results among all the data were achieved by the Convolutional Autoencoder. For the dataset №4 this model also shows the highest values among all the metrics except F1-score. The highest F1-score and precision for the dataset №4 was obtained with the KNN model. We can conclude, that machine learning model KNN and deep learning architectures, such as CNN and LSTM autoencoders were more effective for anomaly detection tasks.

In accordance with the results obtained in section 3 of this work, all models, except OCSVM, showed high results (F1-score – more than 90%) on some individual datasets. This indicates that almost all the approaches studied can be very effective in individual cases, taking into account the specifics of the data.

It should also be noted that some models (KNN, CNN AE, LSTM AE) were able to identify not just point anomalies, but saw an abnormal stage at a certain time interval. At the same time, all models detected the beginning and end of the abnormal stage with errors.

In industry, it is not so critical to define normal data as abnormal as not to notice a real anomaly. From this point of view, the KNN model seems to be on of the most effective, for the dataset №4 model indicated the biggest amount of the true anomalies – 311. It is noteworthy that the CNN AE model did not identify any false anomalies. This can also be interpreted as a positive trait, since false anomalies also require resources and participation in order to understand their nature and determine that they are false.

The direction for further work is the study of various models of deep learning. According to the results of this work, Autoencoders could be powerful in detecting anomalies in the industrial time series. There are some architectures, based on Autoencoders, which could be investigated for explored data, for example Variational Autoencoders. What is more, an analysis of existing research in this area has shown that Generative Adversarial Nets also can be can be successfully applied to the detection of anomalies in time series. In addition to the reconstruction-based and generative models, there are forecasting deep learning models, which could be explored for the investigated problem.

In addition to applying a variety of models, further work may include classification of the main types of disturbances in the water circulation system, which can be determined by the nature of data changes. This will further determine which model is most effective for detecting anomalies in each type of violation.

It can be concluded that machine learning methods are very useful and prospective for the anomaly detection in industrial time series. At the same time, it is important to take into account the specifics of the data and possible scenarios for the occurrence of system failures. Different models can be effective for different scenarios. At the same time, classical machine learning models are simpler and faster, while deep learning architectures require more time and skills for correct implementation, but can be more effective and flexible.

LIST OF USED SOURCES

1) Kevin Feasel. (2022). Finding Ghosts in Your Data: Anomaly Detection Techniques with Examples in Python. Springer. https://doi.org/10.1007/978-1-4842-8870-2.

2) Charu C. Aggarwal (2017). Outlier Analysis. Springer. https://doi.org/10.1007/978-3-319-47578-3

3) Kishan G. Mehrotra, Chilukuri K. Mohan, HuaMing Huang. (2017). Anomaly Detection Principles and Algorithms. Springer Cham. https://doi.org/10.1007/978-3-319-67526-8

4) Zhipeng Chen, Zhang Peng, Xueqiang Zou, and Haoqi Sun: Deep Learning Based Anomaly Detection for Muti-dimensional Time Series: A Survey. W. Lu et al. (Eds.): CNCERT 2021, CCIS 1506, pp. 71–92, 2022. https://doi.org/10.1007/978-981-16-9229-1_5

5) Du, H., Duan, Z. Finder: A novel approach of change point detection for multivariate time series. *Appl Intell* **52**, 2496–2509 (2022). https://doi.org/10.1007/s10489-021-02532-x

6) Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang Y, Yan X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Advances in Neural information processing systems, pp 5243–5253

7) Salinas D, Flunkert V, Gasthaus J, Januschowski T. (2019). Deepar: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting

8) Guo T, Lin T, Antulov-Fantulin N. (2019). Exploring interpretable lstm neural networks over multi-variable data. In: International conference on machine learning, pp 2494–2504

9) Taylor SJ, Letham B. (2018). Forecasting at scale. Am Stat 72(1):37–45

10) Krzysztof J. Cios, Roman W. Swiniarski, Witold Pedrycz, Lukasz A. Kurgan. (2007). Data Mining. Springer New York, NY. https://doi.org/10.1007/978-0-387-36795-8

11) Iurii D. Katser and Vyacheslav O. Kozitsin, “Skoltech Anomaly Benchmark (SKAB).” Kaggle, 2020. Doi: 10.34740/KAGGLE/DSV/1693952

12) Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, “Isolation Forest”, IEEE International Conference on Data Mining. (2008). (ICDM 08)

13) Yahaya, S.W., Langensiepen, C., Lotfi, A. (2019). Anomaly Detection in Activities of Daily Living Using One-Class Support Vector Machine. In: Lotfi, A., Bouchachia, H., Gegov, A., Langensiepen, C., McGinnity, M. (eds) Advances in Computational Intelligence Systems. UKCI 2018. Advances in Intelligent Systems and Computing, vol 840. Springer, Cham. https://doi.org/10.1007/978-3-319-97982-3_30

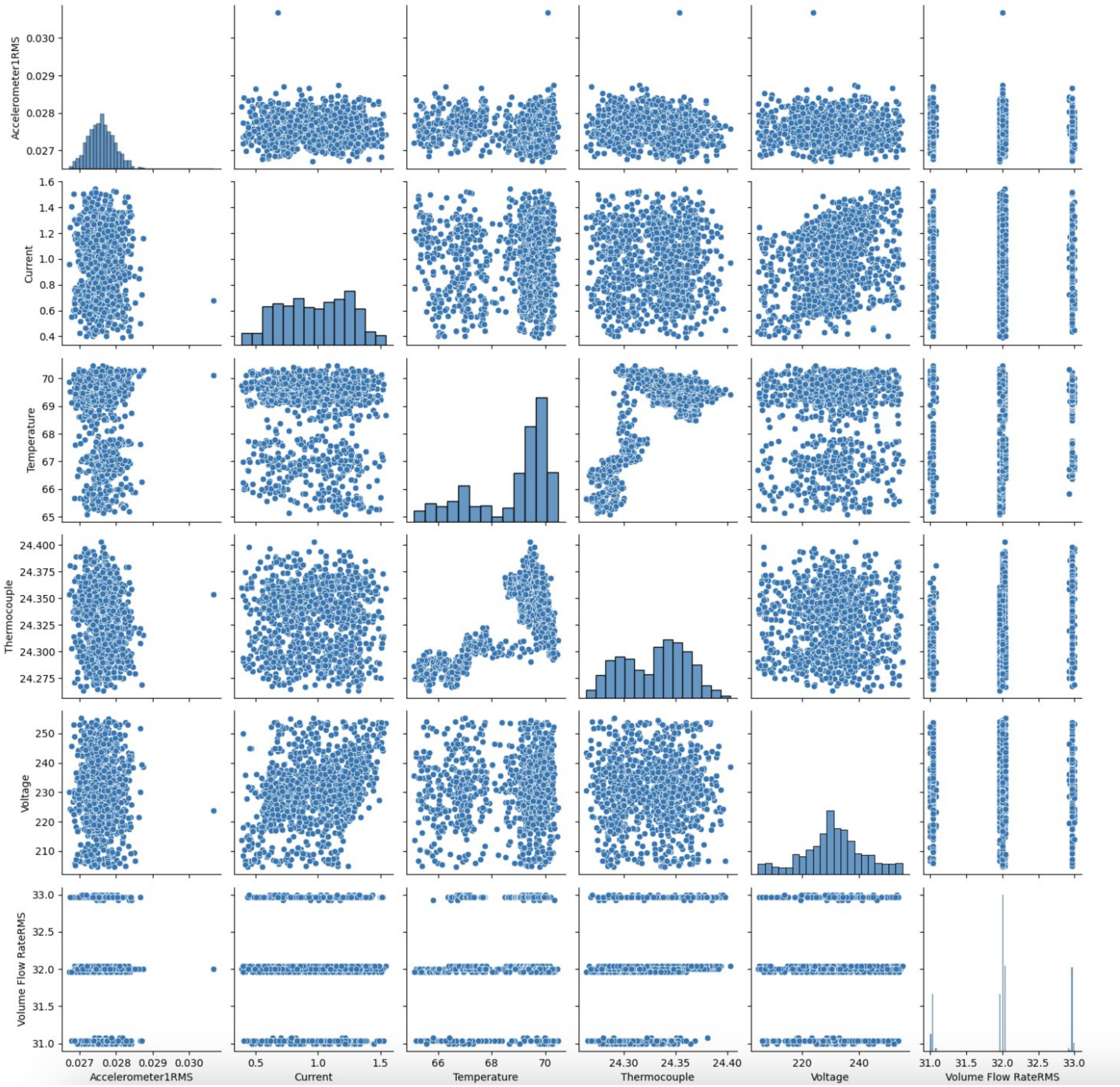
14) Zhao, Y., Nasrullah, Z. and Li, Z., 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of machine learning research (JMLR), 20(96), pp.1-7.

15) Tziolas, Theodoros Papageorgiou, Konstantinos Theodosiou, Theodosis Papageorgiou, Elpiniki Mastos, Theofilos Papadopoulos, Angelos. (2022). Autoencoders for Anomaly Detection in an Industrial Multivariate Time Series Dataset. 10.3390/engproc2022018023

APPLICATIONS

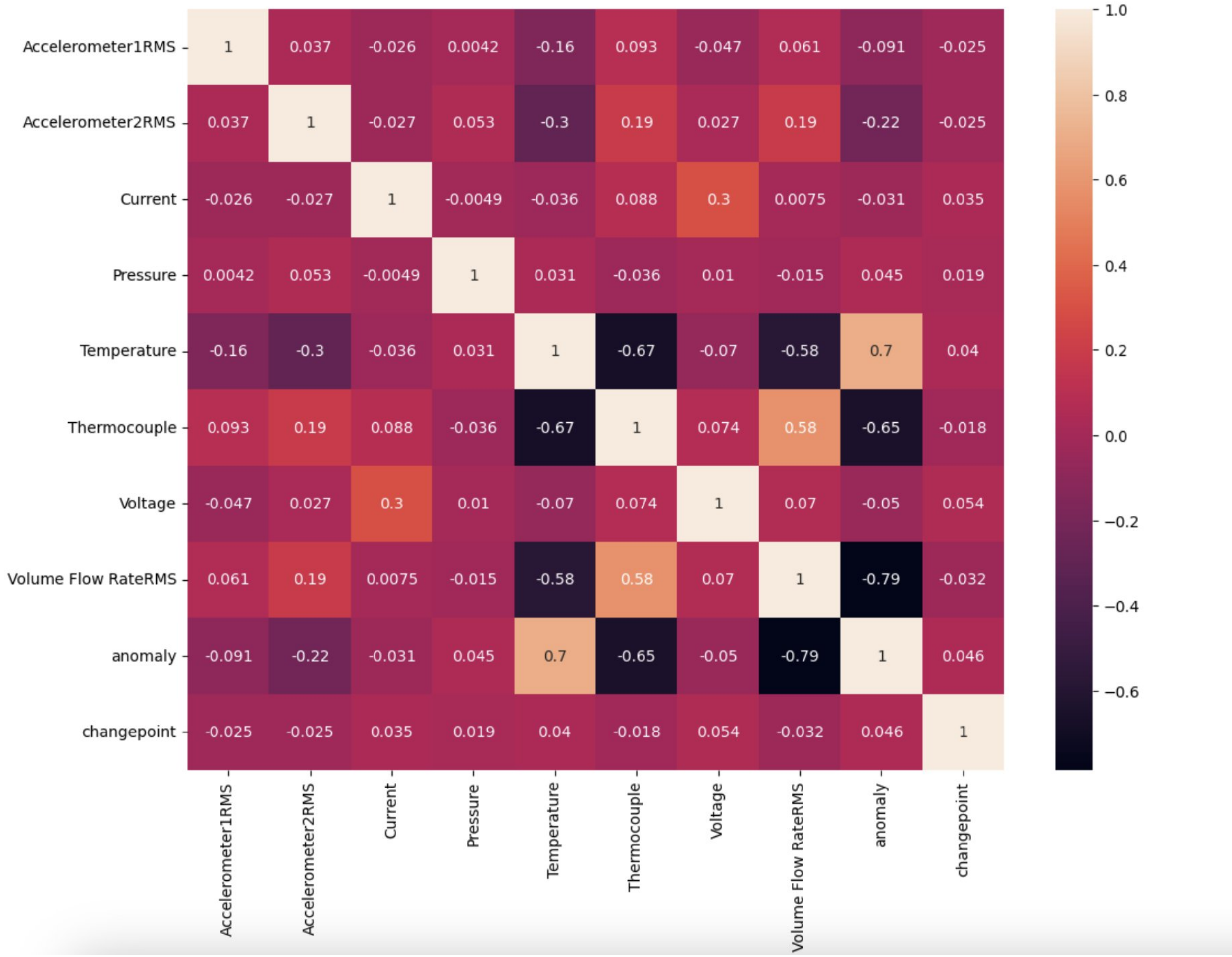
Annex A

The matrix of scatter pair plots



Annex B

The correlation matrix



Annex C

The visualisations of data frames with the best and the worst results (Isolation Forest)

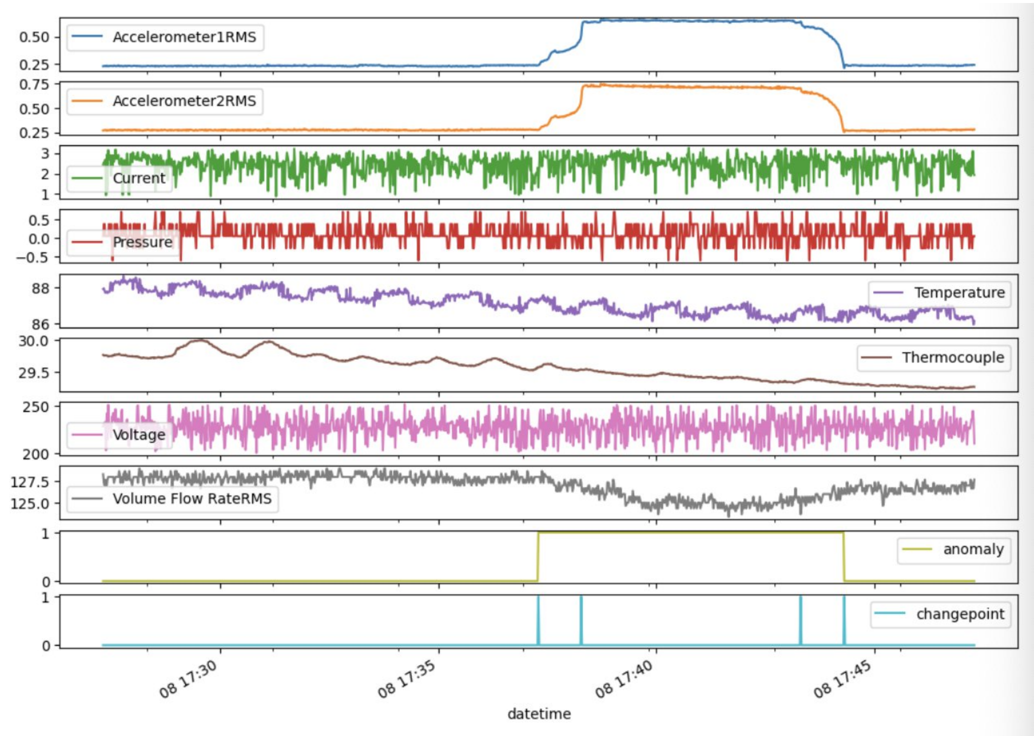


Figure C.1 – The data with the best result

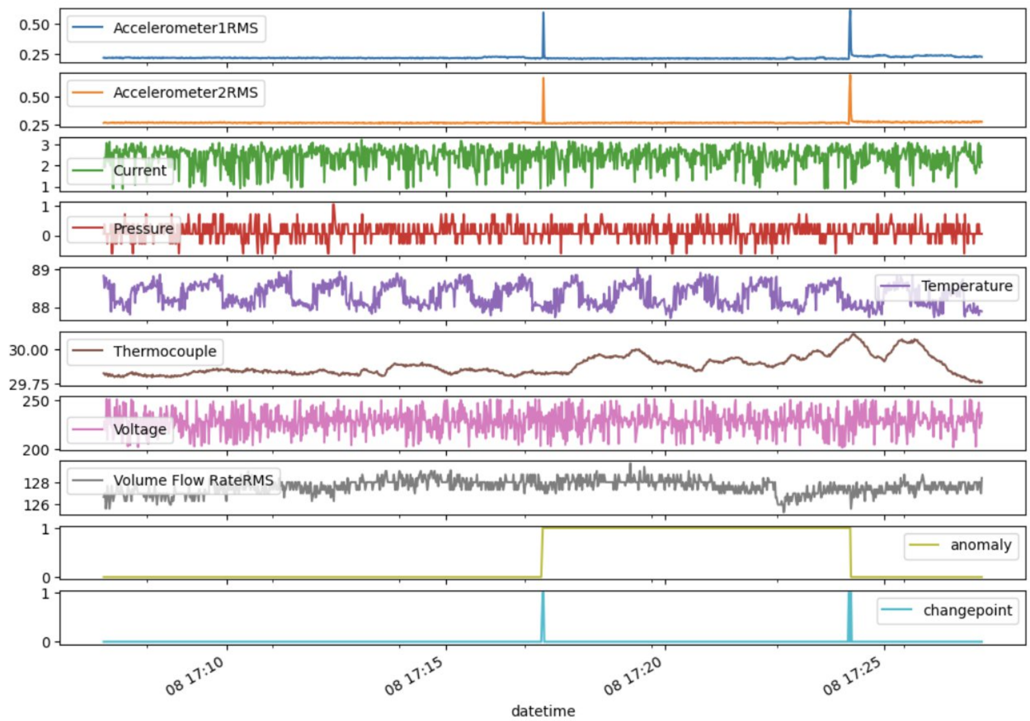


Figure C.2 - The data with the worst results

Annex D

The visualisations of data frames with the best and the worst results (KNN)

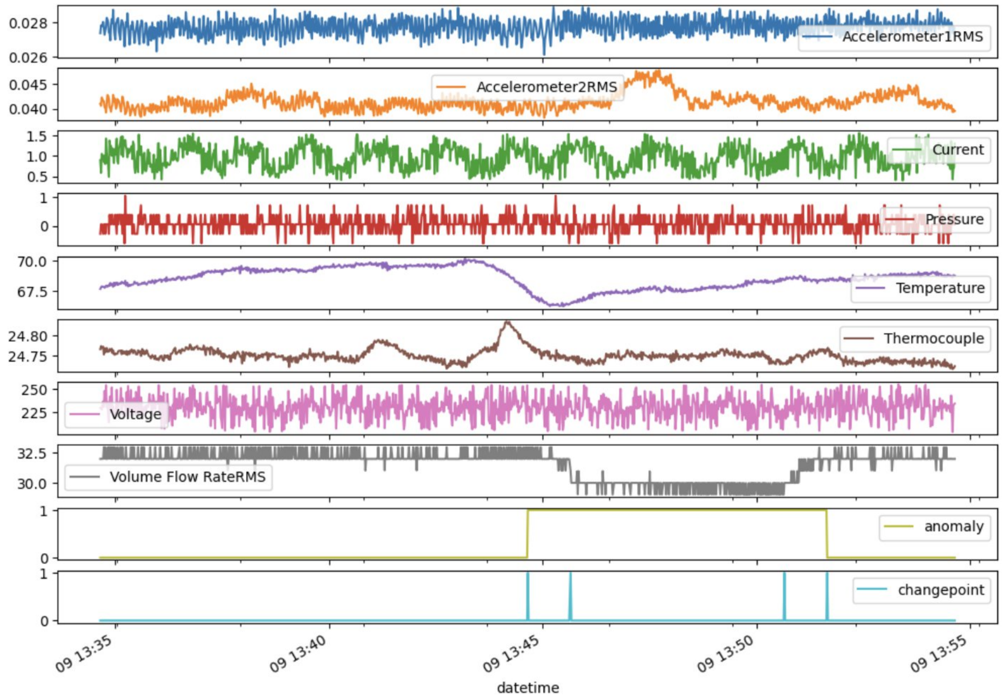


Figure D.1 – The data with the best result

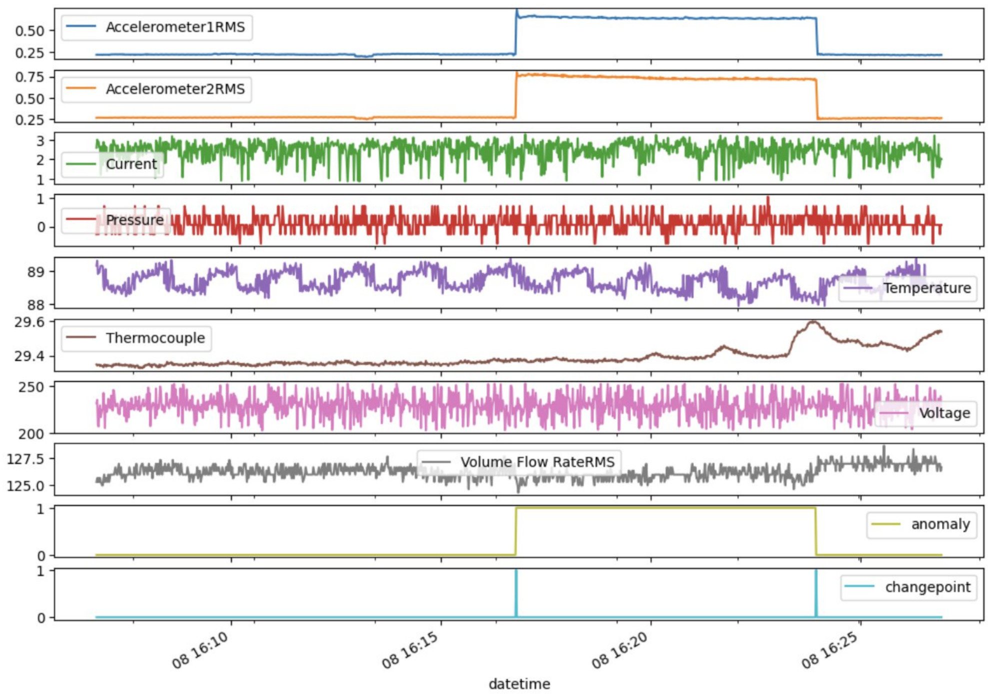


Figure D.2 - The data with the worst result

Annex E

The visualisations of data frames with the best and the worst results (CNN AE)

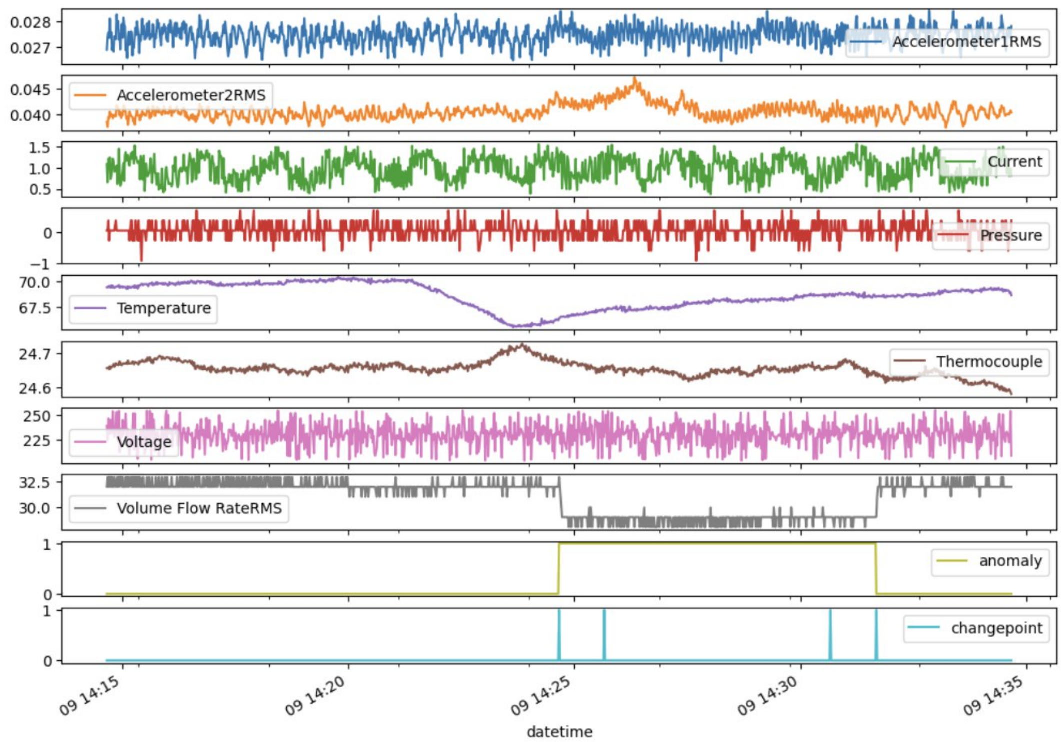


Figure E.1 – The data with the best result

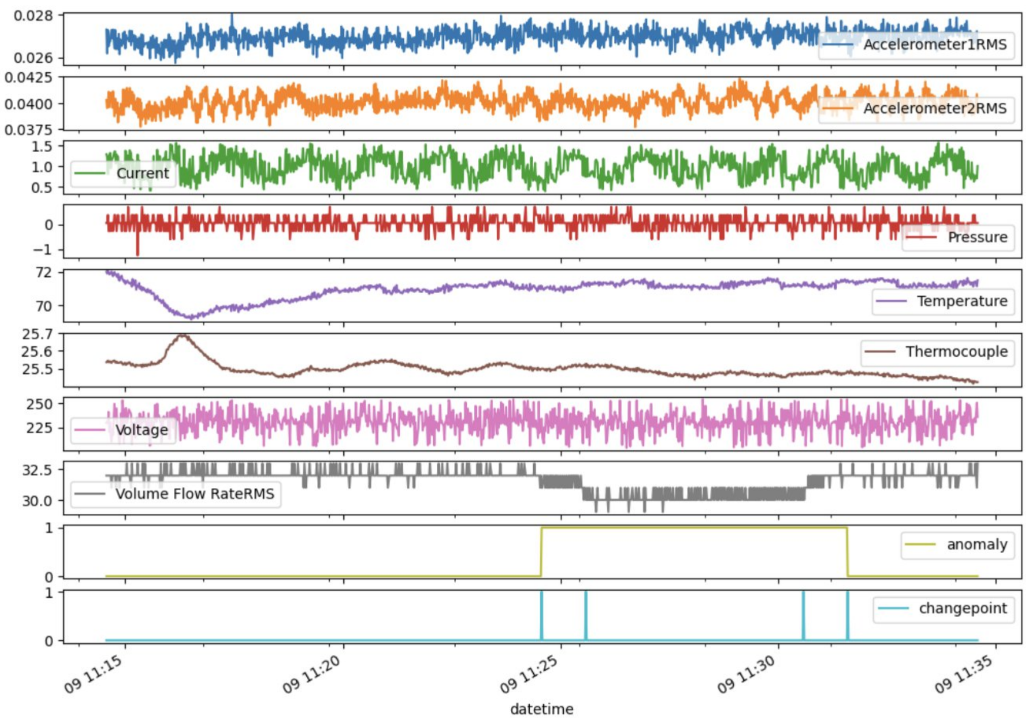


Figure E.2 - The data with the worst result

Annex F

The visualisations of data frames with the best and the worst results (LSTM AE)

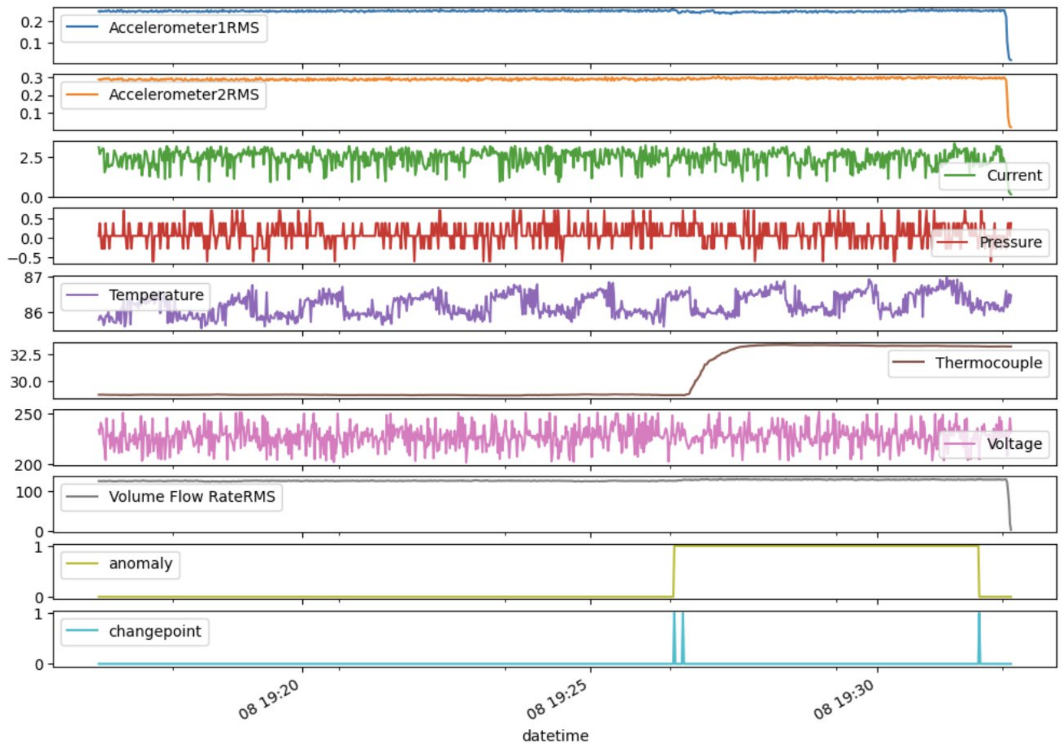


Figure F.1 – The data with the best results

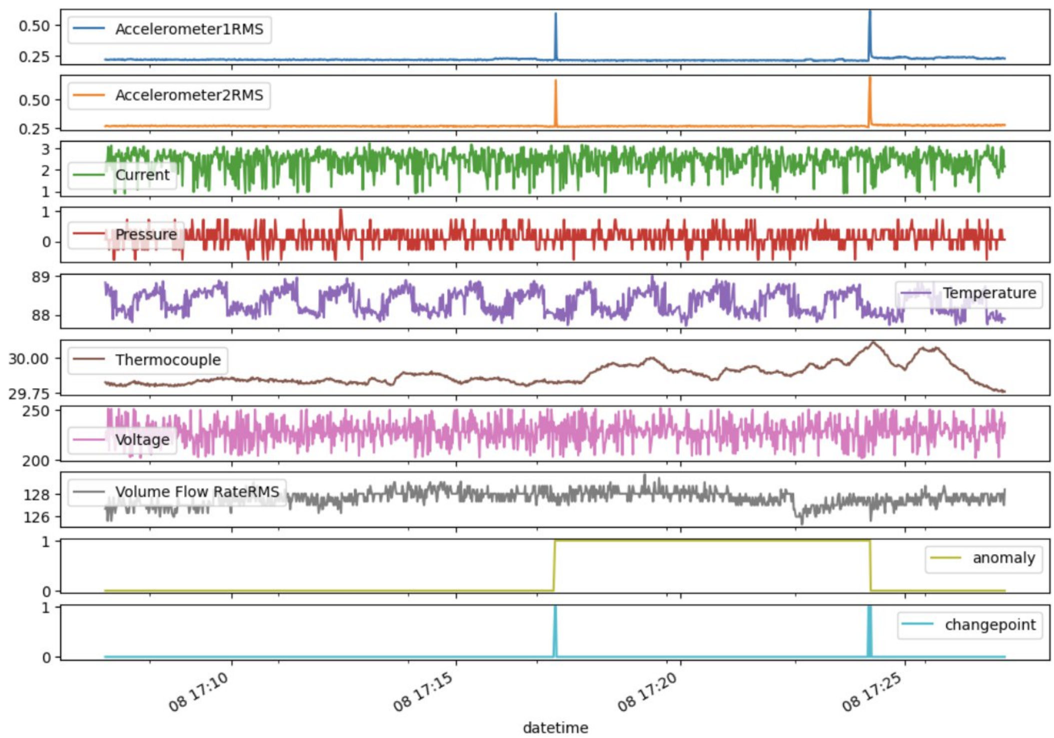


Figure F.2 - The data with the worst result