Vrije Universiteit Amsterdam

Business Analytics

Internship Report

Anomaly detection in Steel Production Lines

Author:
Wenxin Chen
External Supervisor:
Dr. A.R. Griffioen
VU internship supervisor:
Prof. Dr. Koole, G.M. (Ger)
Second Reader:
Msc. Silvestrin, L.P. (Luis Pedro)
Msc. Pantiskas, L. (Leonardos)

Host Organization:
SURF
IJssel
TATA Steel
Facta

January, 2023

Preface

This paper was written for the Master of Business Analytics program at Vrije Universiteit van Amsterdam. It is also a report about an internship I did at SURF as part of the Techport project Use Case B.

I would like to thank Dr. A.R. Griffioen of the Scalable Data Analytics Team at SURF for being my external supervisor. In addition, I would like to thank Wouter Kaag for providing advice and helping in preprocessing the data. Furthermore, I would like to thank all consultants from TATA Steel, Ijssel, and Facta from the Techport network for providing me with knowledge in their fields to assist me before and during my internship.

I would also like to thank Prof. Dr. Koole, G.M.(Ger) for being my supervisor. Lastly, I would like to express my gratitude to MSC. Silvestrin, Luis Pedro and Msc. Pantiskas, Leonardos for being my second reader. They greatly supported me with their advice and guidance.

Abstract

Anomalies in production would result in a significant loss in revenue. Companies are moving towards predictive maintenance to reduce such losses instead of using traditional regular maintenance, which is costly because it is frequent. Noticing the importance of predictive maintenance, the field lab Techport, consisting of a network of companies, universities, and government organizations proposes a project aiming to improve predictive maintenance through data sharing between companies and the use of machine learning techniques. A combination of different sensor data about a particular part (bridle) in the steel production line in TATA Steel is provided by various companies in the network of Techport. With these data and the knowledge provided by various companies in Techport, the goal of this report is three-fold:

- The main goal is to develop an anomaly detection model for the steel production line in TATA Steel using the Techport data.
- Evaluate each data source's usefulness and their combination for the above task.
- The challenges of missing data, and inaccurate and missing labeling exist in this project. Report the influences and suggestions about these challenges.

3 different re-sampling methods and frequencies are used to align and combine the data, resulting in 9 different datasets. These datasets are tested under a baseline model to find the best preprocess method. 4 machine learning algorithms are used: Local Outlier Factor(LOF), One-class Support Vector Machine(OCSVM), Autoencoder and Variational Autoencoder(VAE), where OCSVM is the base-line model.

The report finds the most suitable way to preprocess the data is to take the maximum sensor data in 30 minutes frequency. Using this preprocessed data, the best-performance model is LOF, which achieves a 0.91 F1-score. However, since the anomaly timings provided by Techport are incomplete and inaccurate, a qualitative analysis, as we will see in the following chapters, of whether the model can predict anomalies in advance and whether it can find the anomalies that are not in the labels is essential in this report. In this sense, even though VAE shows a lower F1-score, it successfully finds more anomalies than the others. Hence the report concludes that VAE has the most potential in practice, taking its fastest running speed into account. The report also evaluates each data source and gives suggestions about improving data quality.

Contents

Glossary

IBA	An IT system used by TATA Steel that records the product information(width, length, yield) on the line and the motor operation status.
LOF	Local Outlier Factor.
OCSVM	One-class Support Vector Machine.
PCA	Principle Component Analysis.
RBF	Radial Basis Function.
SAM4	Samotics, a system from the company Samotics that contains sensor data from the company Samotics, which measures the current in the production line.
SVM	Support Vector Machine.
UTW	UptimeWorks, a vibration measurement system from the IJssel company that records various information about the vibration of the steel production line.
VAE	Variational Autoencoder.

1 Introduction

Business Understanding

In steel Production, it is very important to prevent anomalies that can lead to downtime or damage as these would result in significant financial losses. Periodic maintenance can be used, but it is costly. Instead, sensors are used in predictive maintenance to reduce costs beforehand. Effective predictive maintenance can also bring a variety of benefits such as increasing system safety, improving operational reliability and maintenance efficiency, and reducing maintenance, inspection, and repair-induced failures[21]. As a result, the importance of predictive maintenance is rapidly growing. It became one of the major goals of the industry recently and it depends highly on Anomaly detection. Anomaly detection or outlier detection is a broad set of techniques with an aim to identify anomaly patterns that deviate from normal behavior[10]. The importance of predictive maintenance in the world of machine learning can also be reflected by the significant increase[10] in the number of anomaly detection.

Techport project Use Case B

Techport is a so-called field lab in North Holland, a network of universities, ROCs, companies, and government organizations that are committed to the future of the manufacturing industry. "In general, a field lab is a practical environment in which companies and knowledge institutions develop, test, and implement targeted Smart Industry solutions as well as an environment in which people learn to apply these solutions.[3]" Among others, one of the aims of Techport is to develop new prediction models for the steel industry. Techoport Use Case B is one of the projects that aims to analyze and extract insights from the data given by Techport's network. In this project, TATA Steel, Samotics, and IJssel provide the important data and knowledge, and Facta and SURF contribute with knowledge.

Usually, companies only analyze their own sensors and therefore analysis is only limited to a specific one. In the Techport project, it is possible to investigate data from multiple sensors, since companies agreed to share their data. There are, for example, vibration sensors that provide vibration and acceleration measurements, sensors that detect misalignment, power consumption, et cetera. In the end, Techport wants to know whether the combination of sensors and production information can provide improvements in preventive maintenance of the steel production line and optimize the machine's lifetime.

Problem Statement

The Techport project provides sensory and production data about a particular part (called bridle) in the steel production line in TATA steel, where the bridles create tension on the metal strip between them. The data are collected from three different sources, IBA (an IT system from TATA Steel), Up-time-works (a system from the company IJssel with their own sensors), and Samotics (a system from the company Samotics). This report aims to build an anomaly detection model using the Techport data and analyze the usefulness of different data sources and their combination. Different re-sampling and preprocessing techniques are also tested on different models.

Challenges: Data Quality

In the context of big data in predictive maintenance, one open challenge is data acquisition, where the issue lies in obtaining quality data. The collected data is often incomplete, poorly structured, or unannotated[8].

One of the main challenges that exist in real-life is data missing because many data mining and machine learning techniques cannot process incomplete data directly. While discrete missing values can be imputed in many ways, very few studies have attempted to assess the imputation for continuous missing values[14]. Moreover, missing data are often poorly handled and reported, even when adopting advanced machine learning methods for which advanced imputation procedures are available[17]. Having a long period of continuous missing data is especially difficult for most imputation methods to effectively impute.

In data science and machine learning, correct labeling is critical[9]. The first problem is called the problem of the missing labels, where some data are not labeled(i.e. indication of anomaly or normal is only available for a part of the data)[15]. A second problem is incorrect labeling. In the case of anomaly detection, it means the anomaly data is labeled as normal or the other way around.

In this project, whether this kind of challenge exists and how to improve will also be discussed.

Research Question

A considerable number of papers and research have been published on anomaly detection, showing promising results(e.g. [18], [13] and [7]). Nevertheless, most of them are based on ideal datasets, whereas in real-life cases, challenges such as those described in the chapter “Challenges: Data Quality” exist. Taking this into account, the aim of this paper is three-fold.

- Can anomaly detection model(s) be made using the Techport data? This is the main goal of this report.
- During this process, whether the above-mentioned data-related challenges exist in Techport and how do they impact the result? In other words, can

anomalies be detected using the data given by Techport?

- Whether combined data from different data sources have an added value to anomaly detection? If not, analyze the reasons.

Thesis Structure

A total of four anomaly detection algorithms are explored in this thesis to find the best-performing model for detecting anomalies in the data.

The second chapter of this thesis will focus on explaining the project pipeline and the algorithms used. The third chapter will be about the exploration and description of the data, and the cleaning and preprocessing of the data will be discussed in chapter four.

The experimental setups and the evaluation methods will be described in chapter 5. Chapters 6 and 7 will present the results and evaluations. Lastly, chapter 8 will discuss suggestions and future works.

2 Methodology

This chapter will focus on the project pipeline and the relevant theories and literature.

2.1 Project Pipeline

The project is implemented according to the pipeline shown in Figure 1, starting with data wrangling and cleaning. Unlike in ideal situations, where the pipeline is mostly linear, the exploration, preprocess and model steps would cycle until a satisfying result is obtained. During the cycle, new challenges and problems would arise, leading to changes in the strategies.

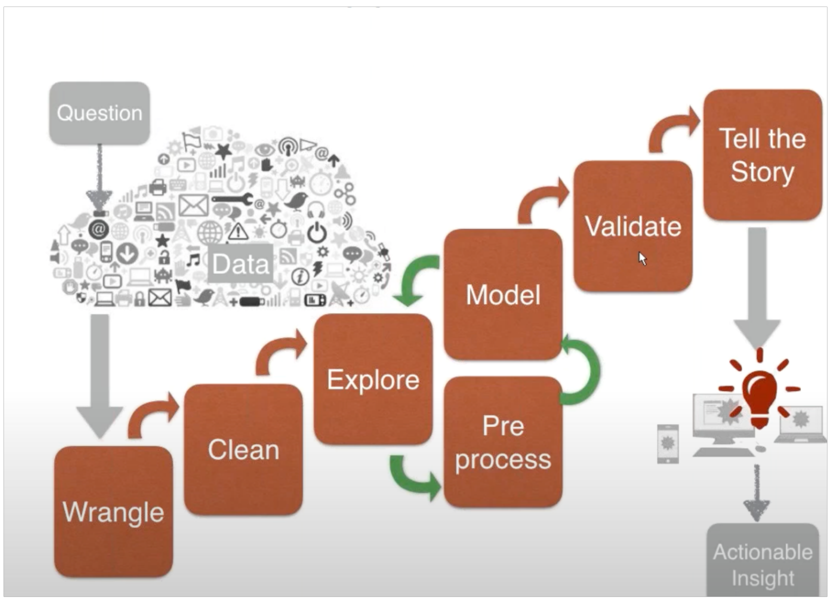


Figure 1: Project Pipeline Indication [1]

2.2 Local Outlier Factor

LOF(local outlier factor) is an algorithm proposed by Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. It is commonly used in anomaly detection by finding anomalous data points by measuring the density of the data point with respect to its neighbors, and comparing it with its other neighbours[5].

LOF is based on measuring the distance between data points within a set of data points. Let $k_{dist}(A)$ (k-distance of A) be the largest distance of the data point A to its k closest neighbors. This set of neighbour with this $k_{dist}(A)$ is called k-distance neighborhood (k_{dist_nh}). These are used to define the concept of reachability distance.

Rechability Distance The reachability distance of A to a certain point B($k_{reach\_dist}(A, B)$) is defined as the maximum between the distance of these two points($d(A, B)$) and the k-distance of B. Formally written as:

$$k_{reach\_dist}(A, B) = max(k_{dist}(B), d(A, B))$$

A visual representation of reachability distance can be found in figure 2.

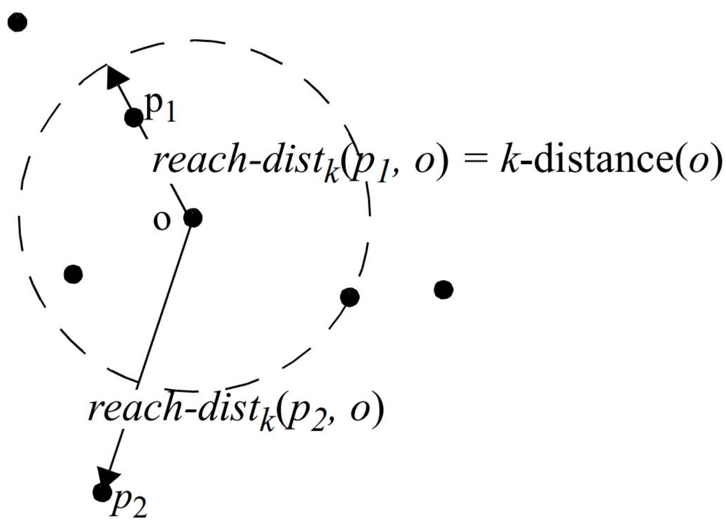


Figure 2: reach-dist(p1,o) and reach-dist(p2,o), for k=4[5]

Local Reachability Distance Local reachability distance of A ($k_{lrd}(A)$) is the inverse of the average reachability distance of the A with respect to its k-distance neighborhood. Formally written as:

$$k_{lrd}(A) = \frac{1}{\left(\frac{\sum_{B \in k_{dist\_nh}(A)} k_{reach\_dist}(A,B)}{|k_{dist\_nh}(A)|}\right)}$$

Local Outlier Factor As this is where the algorithm got its name from, this is the degree that measures the degree of a data point being an outlier. The local reachability density of the data point is compared with its neighbors using this formula:

$$k_{lof}(A) = \frac{\sum_{B \in k_{dist\_nh}(A)} \frac{k_{lrd}(B)}{k_{lrd}(A)}}{|k_{dist\_nh}(A)|}$$

This is the average local reachability distance of the k-distance neighborhood of A divided by the local reachability distance of A. Hence, if LOF is approximately equal to 1, then A is not an outlier. A LOF value significantly larger than 1 indicates an outlier, while below 1 shows a higher-density region.

2.3 One-class SVM

In the world of predictive maintenance, Support Vector Machine (SVM) in a commonly used algorithm[6]. Support Vector Machine (SVM) is a very powerful non-parametric binary classification predictive method to solve regression and classification problems. It is based on statistical learning theory in machine learning[22], with a primary concept of reducing generalization error in classifying and detecting to achieve predictive accuracy, as proposed by Vapnik[23]. Support Vector Machine aims to separate classes in the optimal way possible by choosing the hyperplane(s) in which the nearest data points from each class are maximized. This is called the max-margin, and the data points that help determine the hyperplane are called support vectors.

One-class SVM(OCSVM) is a special unsupervised method proposed by Muller[16]. It can be used in outlier or anomaly detection problems by finding a kernel-based classifier with maximum margin from the origin in the feature space[16] [20]. Figure 3 shows an indication of how OCSVM works intuitively. The red circle in the figure represents the decision boundary or "frontier" learned by the algorithm, which serves to separate the data points into two classes. Data points located outside of this circle can be considered outliers.

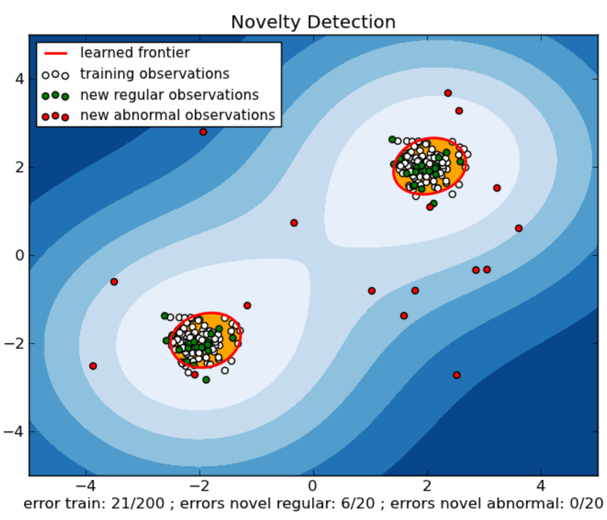


Figure 3: An indication of OCSVM[2]

The training data is projected into a higher-dimension feature space using a kernel $k(x, y)$. This kernel can be set to 'RBF' (Radial Basis Function), which computes the similarity of x and y using the following equation, where the hyper-parameter δ is the variance

$$k(x, y) = exp(-\frac{\|x - y\|^2}{2\delta^2})$$

After projection, data is separated from the origin by creating a hyperplane. Outliers are the data that the hyperplane cannot separate. This is done by solving the following quadratic equation[19]:

$$\min_{\omega,\xi,\rho} \frac{1}{2} \|\omega\|^2 + \frac{1}{\nu\ell} \sum_i \xi_i - \rho \tag{1}$$
$$s.t.(\omega \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0$$

In this equation, $i \in \ell$, and ℓ is the number of instances. ξ_i is a nonzero slack variable. $\nu \in (0, 1]$ is the trade-off between the number of outliers the algorithm is allowed and the number of support vectors used. This is one of the most important hyper-parameter to tune when training.

After solving equation (1), the following equation can be used to classify new incoming data under the trained model:

$$f(x) - sgn((\omega \cdot \Phi(x)) - \rho)$$

2.4 Auto-encoder

Autoencoder is a type of unsupervised artificial neural network. A representation for the data is learned(encoding) by training the network to ignore noises by dimensional reduction[12].

An autoencoder consists of two main parts: the decoder and the encoder. The decoder maps the input data x to the feature space z. The decoder (f) maps the abstract feature z back to the original space to obtain the reconstructed data x'(shown in figure 4). To optimize the encoder and the decoder, the model aims to minimize the reconstruction error, which is the error between the input x and the reconstructed x' using a certain error function L:

$$f, g = argmin_{f,g} L(x, f(g(x)))$$

For neural network-based AutoEncoder, the encoder compresses the data by reducing the number of neurons layer by layer. The decoder then increases the number of neurons back to reconstruct the input data.

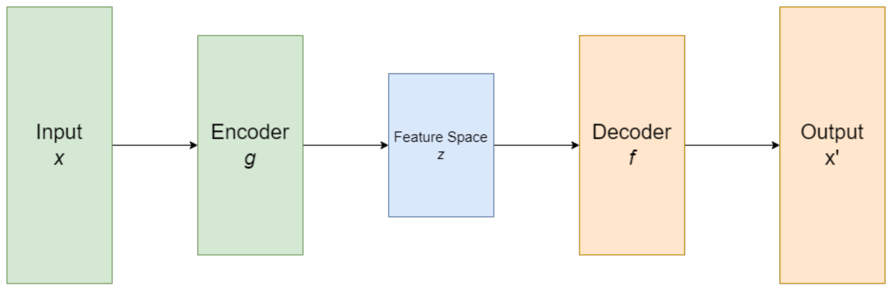


Figure 4: An indication of the autoencoder

2.5 Variational Autoencoder

Apart from Autoencoder, Variational AutoEncoder (VAE) can also be used in anomaly detection, and sometimes may outperform the traditional Autoencoder[4]. Variational AutoEncoder (VAE) was proposed by Kingma et al. in 2014. Different from the autoencoder, which maps the input x to a fixed abstract feature z, VAE assumes that the abstract feature z of the sample x obeys a normal distribution $N(\mu, \sigma^2)$. The abstract feature z is then sampled from this distribution. Finally, the decoder reconstructs the input data based on z[11].

The distribution of the input data X is constructed by the following formula:

$$p(X) = \int_z p(X|Z)p(Z)$$

$p(X|Z)$ is a model that generates X from Z, where it is assumed that Z follows a standard normal distribution, i.e. $p(Z) = N(0, I)$. To maintain the one-to-one correspondence relation between the original data point X_k and the Z_k after decoded, VAE configures an exclusive normal distribution for each of the original samples x_k by fitting it into the neural network. Random samples are then taken from these normal distributions, and the decoder map these samples back to the original shape to reconstruct the data. The optimization process is the same as described in chapter 2.4. An indication of VAE is shown in figure 5.

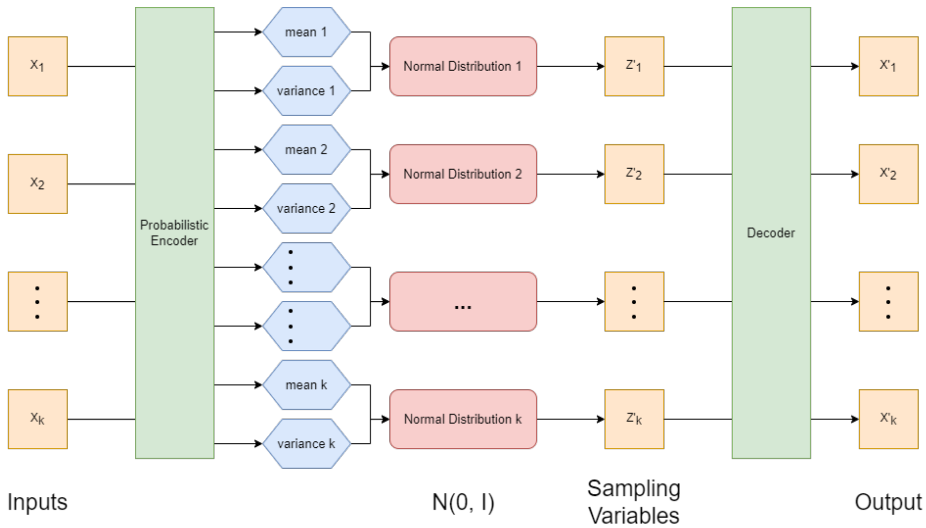


Figure 5: An indication of VAE

3 Data Overview and Exploration

The goal of Techport Project Use Case B is to detect anomalies in the production line(Figure 6). Loopers and bridles are part of the component of the production line.

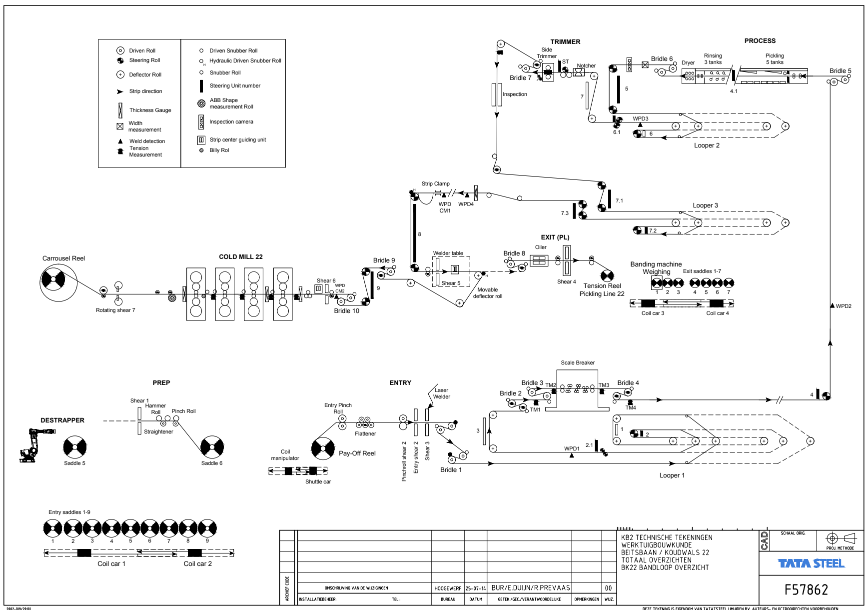


Figure 6: An indication of the production line[2]

The sensor and production data of bridle2, bridle 5, bridle 6, and 7 provided by Techport are from 3 main sources: IBA(a data acquisition system used in TATA Steel), UTW(UptimeWorks, the vibration measurement system) and SAM4(System of samotics, measuring the current). In addition, an incident report that records the approximate date of reported faults by the employees of Techport is given as an indicator to test the results. After a first analysis, this report decides to focus on Bridle 5 and 6 because they provide the most complete data.

Data from different sources have different time spans, time frequencies, data sizes, and data quality. Table 2 presents some basic information about the data of Bridle 6.2 from each source as an example(others in Appendix A). The next few chapters will include some more detailed information about the data.

	time span	Entries	frequency	Features
IBA2020	20-01-01 to 21-01-01	3162241	10 seconds	8
IBA2021	21-01-01 to 21-06-10	1386112	10 seconds	8
IBA2022	21-07-02 to 22-07-08	109618/11320/5859	re-sampled to 3/30/60 minutes	8
IBA Fault&Alarm code	20-11-08 to 21-12-31	2616032	10 seconds	label
UTW	20-09-30 to 22-05-02	132800	irregular	59
SAM4	19-12-31 to 22-05-02	1648361	irregular	13

Table 2: Basic information about the three datasets on Bridle 6.2

3.1 IBA

The IBA data comes from the internal IT system of TATA Steel. It contains the product information(width, length, yield) on the line and the motor operation status. The two and half years of data are given separately in 3 parts: IBA2020, IBA2021, and IBA2022. In IBA data, all features and all bridles have the same percentages of missing values. Furthermore, the missing data appears at the same time, which makes IBA easier to pre-process. After aligning the three datasets, the missing percentages of bridle 6.2 are shown in table 3 (other bridles also have the same missing percentages). Figure 7 shows an example of how the IBA data looks like(confidential data not included).

Feature names(simplified)	Missing percentages	Feature type
Width	3.9%	Product information
Thickness	3.9%	
Yield	3.9%	
Motor Tempreture	3.9%	Motor Status
BRIDLE 6_2_V_act_m_s	3.9%	
BRIDLE 6_2_N_act_rpm	3.9%	
BRIDLE 6_2_TQ_act_Percent	3.9%	

Table 3: Feature missing percentages Bridle 6.2 IBA

3.2 SAM4

SAM4 contains sensor data from the company Samotics, which measures the current. Although it contains irregular time frequencies, most of them are within 1 minute. The missing data percentages in SAM4 show a large difference in each feature. Table 4 shows the missing percentages of each measurement. Although most of the missing values are over 25%, this is because the time frequencies that SAM4 collects the data are mostly in milliseconds. Hence the number of missing values can be improved by resampling to a larger time-frequency (e.g. 3 minutes). Figure 7 shows an example of the SAM4 raw data.

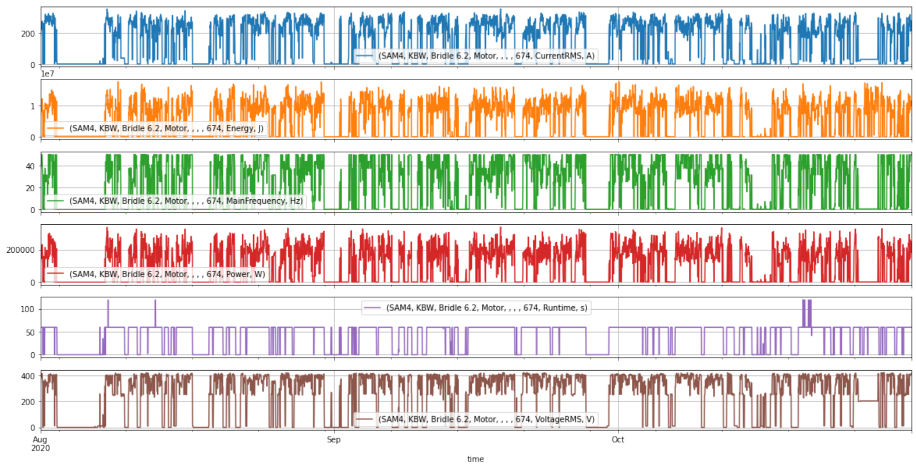


Figure 7: SAM4 raw data on August, September, October 2020

Feature Name	Missing percentages
CurrentRMS	27.6
Energy	27.6
HVFLoss	93.8
MainFrequency	27.6
MotorAssetLoss	93.8
MotorOperationalLoss	95.8
Power	27.6
PowerFactor	47.2
Runtime	27.5
StartStops	99.2
UnbalanceLoss	93.8
VoltageL2LRMS	57.4
VlotageRMS	27.6

Table 4: Missing Percentages SAM4

3.3 UTW

The UTW (UptimeWorks) is a vibration measurement system from the IJssel company that records various information about the vibration of the production line. As shown in table 2, UTW has significantly fewer entries compared to other raw data. This is due to its large amount of missing data time period and irregular time frequency to collect the data. The UTW data have 772 different time frequencies, spreading from the shortest of 9 seconds to as long as 35 days

5 and a half hours, which means that UTW did not collect any data for over a month, and this is not a single case. There are 5 long continuous missing cases: 35 days, 27 days, 25 days, 17 days, and 14 days. Even within one day, the irregular time frequency can differ from 9 seconds to 19 hours. As a result, although UTW has a low missing percentage on its own, it would have a very large proportion of missing values when aligning with IBA and SAM4. Figure 8 shows the missing data distribution of the UTW data after aligning, where blue indicates missing data.

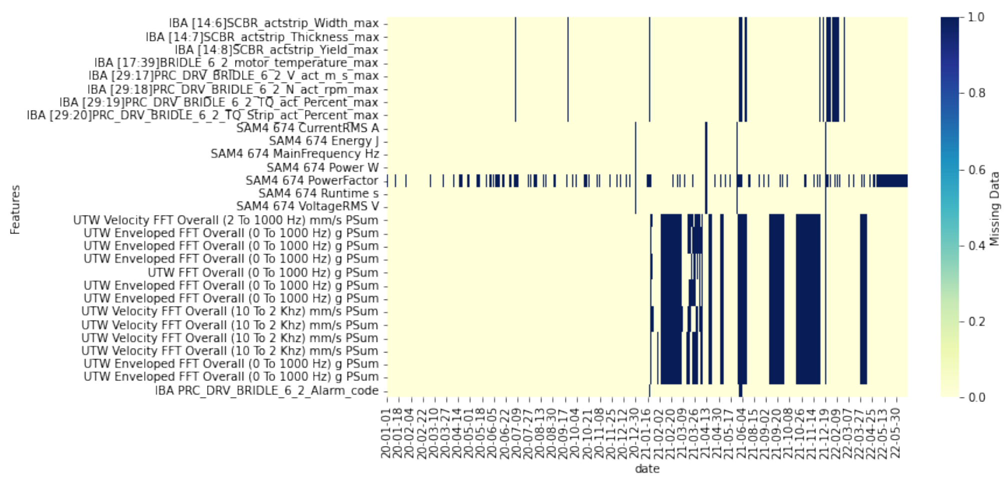


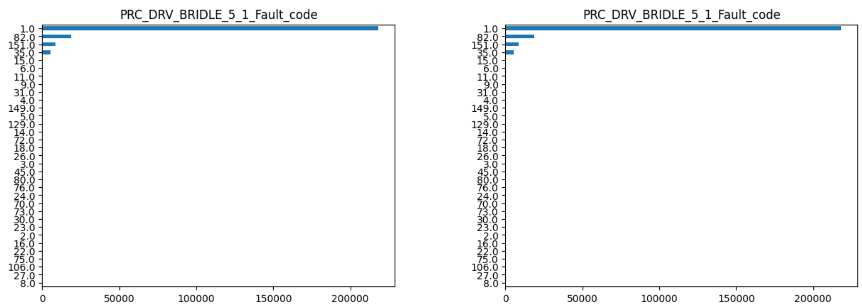
Figure 8: Missing data indication: blue indicates missing, yellow indicates data

3.4 Fault and Alarm codes

The IBA Fault and Alarm codes are a set of indicators that are used to communicate potential anomalies or issues within the IBA system. These codes are typically presented separately from other data within the system and are intended to serve as a warning or alert to system operators or maintenance personnel Each alarm and fault code is indicated by a code number. It is important to note that an alarm code does not necessarily mean that there is an actual anomaly. And vice versa, when there is no alarm or fault code, it does not necessarily mean that the system is normally operating. The codes only serve as an approximate indication of anomalies.

In Bridle 5, there are only fault codes exist, but no alarm code. Figure 11 shows the frequency of each fault code in Bridle 5.1 and 5.2. It can be observed that both sub-bridles have one dominating fault code.

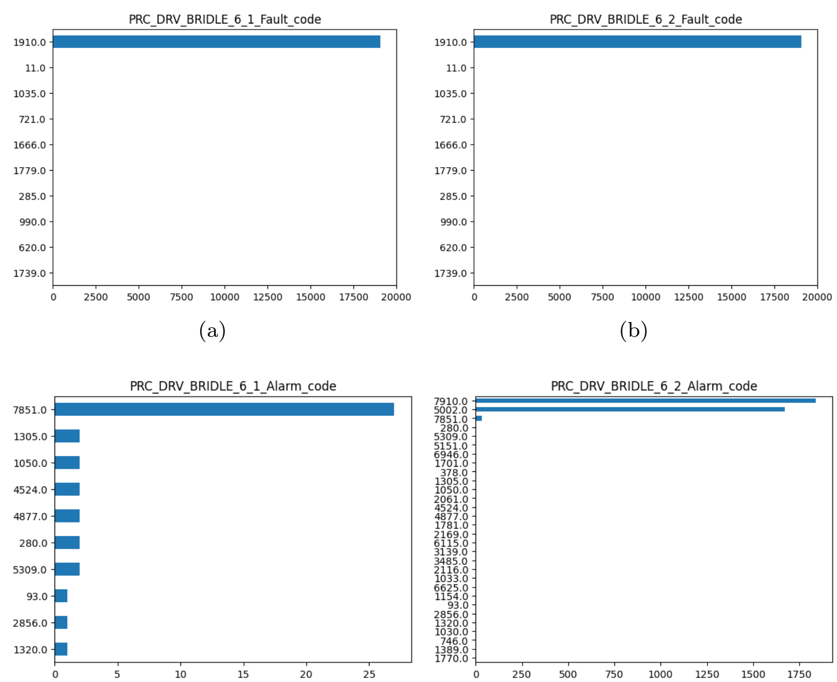
This is also the case for the alarm and fault codes in Bridle 6, shown in figure 10. Both sub-bridles have one fault code that has a dominant amount, with the alarm codes exhibiting more variability comparably. Still, some alarm codes displayed a significantly higher frequency. This pattern served as the basis for



(a) (b)

Figure 9: Fault code Frequency Bridle 5.1 (a) 5.2 (b)

applying the subsequent binomial transformation.



(a) (b) (c) (d)

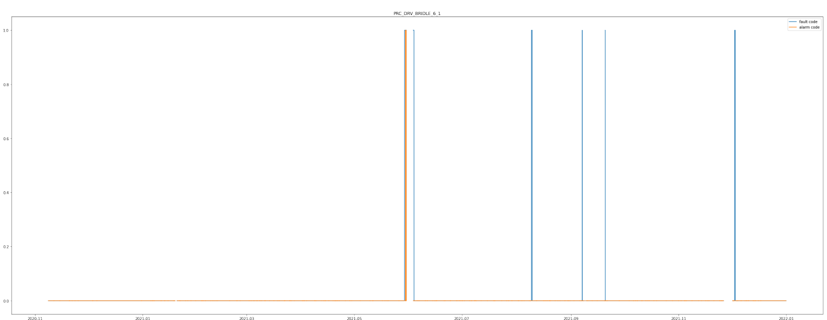
Figure 10: Frequencies for fault codes of Bridle 6.1 (a), 6.2 (b). Frequencies for alarm codes of Bridle 6.1 (c), 6.2 (d).

The exact amount of anomalies compared to normal are shown in table 5, where all alarm codes are transformed to binary.

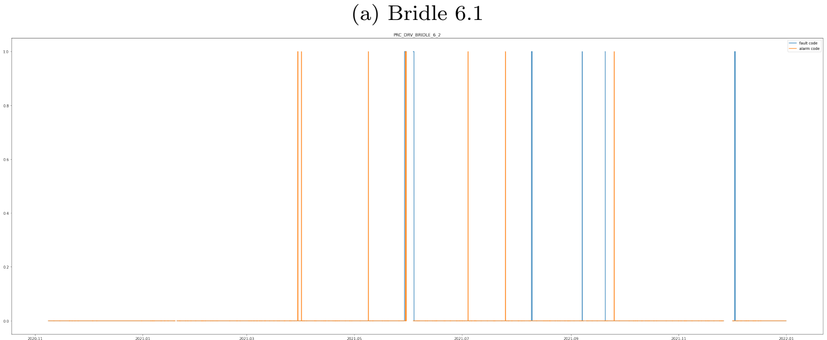
	Normal	**Anomaly**
PRC_DRV_BRIDLE_5_1_Fault_code	3262342	251322
PRC_DRV_BRIDLE_5_2_Fault_code	3462556	51108
PRC_DRV_BRIDLE_6_1_Fault_code	3494596	19068
PRC_DRV_BRIDLE_6_2_Fault_code	3494596	19068
PRC_DRV_BRIDLE_5_1_Alarm_code	3513664	None
PRC_DRV_BRIDLE_5_2_Alarm_code	3513664	None
PRC_DRV_BRIDLE_6_1_Alarm_code	3513622	42
PRC_DRV_BRIDLE_6_2_Alarm_code	3510077	3587

Table 5: Exact amount of fault and alarm codes

Table 5 shows that fault codes appear much more frequently than alarm codes. However, an alarm code is not exactly preceded by a fault code, and vice versa, which means that they are not necessarily related to each other. This can be observed in figure 11, where the orange lines indicate alarm codes and blue lines indicate the fault codes. It can also be observed that although table 5 shows a large number of fault codes, there are very few blue lines in the figure. This is because the fault codes are concentrated on specific dates.



(a) Bridle 6.1



(b) Bridle 6.2

Figure 11: Fault and alarm codes appearances Bridle 6.1 (a) 6.2 (b)

3.5 Incident report

The incident report is a communication record between IJssel and TATA Steel. It summarizes the approximate dates and time span of the damages that have been communicated by TATA and IJssel. It is noticeable that not all damages are recorded in this spreadsheet, and the dates and time span solely depend on the combination of email communication, the time when the employees report the damages to the system, and the memory from the employees, hence can only be used as an approximate of the model.

3.6 Data quality

To conclude, although Techport managed to provide various types of data from different companies, the quality of the data is still problematic. It is severely impacted by long continuous periods of missing data, non-standardized data collection, and inaccurate indications of anomalies. Apart from the issue of missing data, data from UTW is collected in an irregular time stamp, which makes it hard to merge and align with other data. Another main challenge lies in the lack of damage indication. Not all damages are recorded, as well as the exact timing when the damages took place, which makes it difficult to validate and adjust the model.

4 Data Cleaning and preprocessing

This chapter will focus on the cleaning and preprocessing of data. With different preprocessing methods, several different datasets are constructed to be experimented on.

4.1 Data Cleaning

As shown in table 3 and discussed in chapter 3.1, IBA is the most complete data source with the lowest missing rate and the most regulated time-frequency (collected every 10 seconds) of data collection. Hence, its timestamps serve as a baseline to align and analyze other data. At first, data from UTW and SAM4 are aligned with IBA (10 seconds collection frequency). At this step, the python functionality Merge_asof is used. It uses the timestamp of the first dataset as a baseline and finds the nearest timestamps within a certain threshold (2 days) and data of the other dataset. After doing so, the problem of long continuous periods of missing data in the UTW dataset is discovered (shown in Figure 8), which results in a 66% missing percentage. As for SAM4, apart from the features with more than 90% missing that we decided to be dropped from the dataset, the missing rate of the other features also reached more than 20%. To solve this problem, the baseline data IBA is down-sampled to several different lower time frequencies. After down-sampled to 3 minutes, the missing percentage of SAM4 is reduced to 7%, which is acceptable according to expert opinion.

However, it is found that even though the data is down-sampled to 60 minutes frequency, UTW data still reaches a 59% missing rate due to the long continuous missing data period. On one hand, Down-sampling to 60 minutes frequency already reduces the size combined of the dataset from 4548353 entries to 18695 entries. Hence, down-sampling to an even lower time-frequency (for example days or weeks) will lose too much information, hence reducing the missing percentage of UTW data by down-sampling is not an option. On the other hand, simply increasing the threshold to fill the data when aligning is also not an option for the same reason.

After discussing with experts from TATA Steel, it is known that the product production plan, which influences the sensor data most, is irregular. And also due to the lack of data from previous years to compare, we decided against using imputation in UTW data with similar historical data. Imputing using deep learning might be an option. However, as discussed in the Challenges chapter in chapter 1, this is one of the main challenges to be solved and it is out of the scope of this report. Furthermore, most interpolation methods from these are made for completely randomly missing data, while the missing data in the UTW data is continuous and is missing due to possible anomaly (that's not in the labeling), regular repair, and sensor failures. Hence imputing the missing data may result in inaccurate results. This is also proved in follow-up experiments. Of course, there may be an ideal way to impute UTW data, but this is beyond the scope of this report, and we would like to leave this to future works.

4.2 Data preprocess

4.2.1 Features

The datasets for IBA 2020 and 2021 are initially provided as a single, comprehensive file containing raw data, while the dataset for IBA 2022 is divided into separate files and made available through the IBA system. As a result, the IBA 2022 dataset is not initially provided alongside the IBA 2020 and 2021 datasets, making it more challenging to transfer in the same manner as the earlier datasets. With the help from employers in IJssel, IBA 2022 is given as a pre-processed file, down-sampling from the original 10 seconds to 3 minutes/30 minutes/60 minutes. Hence the minimum time frequency to align the data is 3 minutes.

To reduce noise and running time, three down-sampling methods are used when down-sampling: mean, maximum, and minimum, resulting in 9 different preprocessed datasets for bridles 5 and 6 according to the down-sampling method and time. Each of them is experimented with to find the best result.
Data from IBA and SAM4 are combined for each sub-bridles (bridle 5.1, 5.2, 6.1, and 6.2). They are also merged as complete datasets of bridles 5 and 6. The report experimented with the performance of the anomaly detection model on each sub-bridles as well as the complete bridles to find the best result. Figure 12 shows how the datasets are constructed.

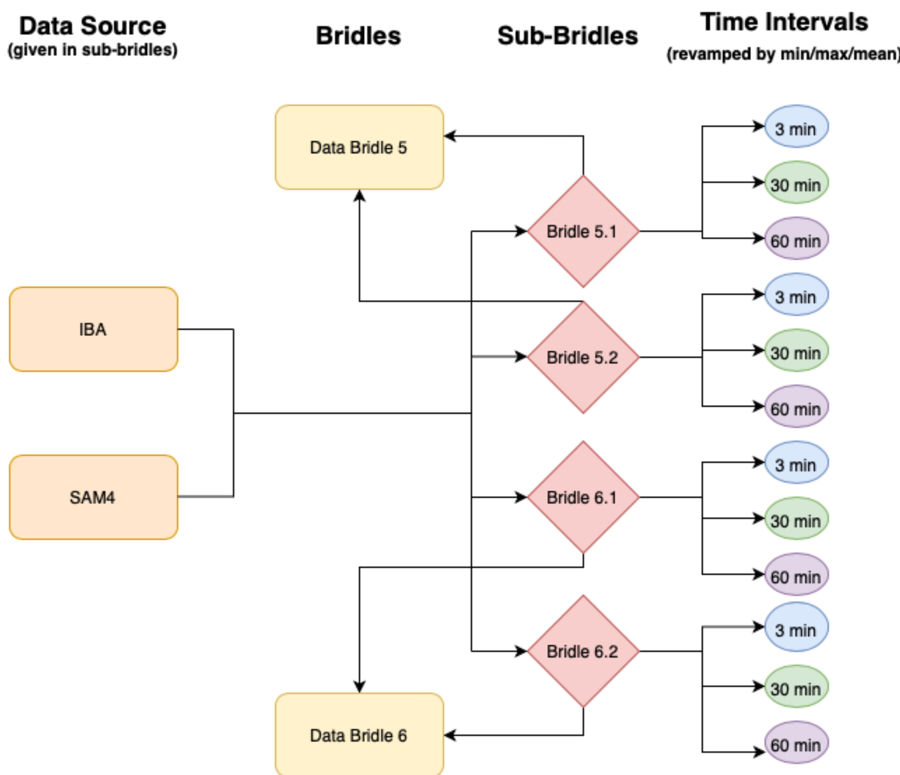


Figure 12: Data Formation

After dropping the columns with more than 20% missing rate, the missing percentages of all the features in the merged dataset are less than 10%. The remaining features include 13 features from IBA and 12 features from SAM4. All the records with missing values are finally decided to be dropped. Interpolating has also been experimented with, but resulting in undesirable predictions.

4.2.2 Labels

After down-sampling, the fault and alarm codes are transferred to binary to use as labels. They are combined to form the labels for the model. The instance or row is marked as an anomaly if the fault code or the alarm code shows an anomaly. Table 6 shows the total number of alarms on the dates they happened and where they happened, and table 7 shows the anomaly dates and bridle shown on the incident report. If the table shows Bridle 6, it means that the anomaly is found in both sub-bridles. It is worth noticing that these two anomaly dates are not overlapping.

Table 6: Dates Fault/Alarm Code Bridle 6

	Fault/Alarm code number	Bridle
2021-05-09	3.0	Bridle 6.2
2021-05-29	22.0	Bridle 6
2021-05-30	28.0	Bridle 6
2021-06-03	18.0	Bridle 6
2021-08-09	21.0	Bridle 6.1
2021-09-07	2.0	Bridle 6
2021-09-25	4.0	Bridle 6.2
2021-12-02	4.0	Bridle 6
2021-12-03	3.0	Bridle 6

Table 7: Dates Incident Report

Incident Report Date	Bridle
18/07/2020	Bridle 6
04/01/2021	Bridle 6.1
16/11/2021	Bridle 6
04/02/2022	Bridle 6.2
12/05/2022	Bridle 6

5 Experimental Setup

This chapter will focus on train-test-splitting and the setup of different models.

5.1 Data split and scale

This is a time series data, hence the training set consists of all data from the year 2020 and the first 3 months (January, February, and March) of 2021. April 2021 is used as validation, and the rest of the data (May 2021 to 4^{th} May 2022) is the test set. In addition, this split is used in order to evaluate how a trained model using data from a certain period would generalize to new data in the future. Since the objective of the model is to learn the normal data pattern of the production line. All data labeled as anomalies are moved from the training and validation set. Furthermore, the damage dates recorded in the incident report and all the data collected the week leading up to them are also removed from the training and validation set. This decision is made according to experts' opinions in TATA, and the consideration of the delay report. Thus, the training and validation set is assumed to contain only normal data. This set contains 20978 entries, and the validation set contains 1284 entries.

The test set is split into a normal test set and an anomaly test set. The anomaly test set contains all the data labeled as anomalies from May 2021 to 4^{th} May 2022, and the week leading up to them and including the anomaly date recorded in the incident report. This set contains 516 entries (30 minutes frequency). The rest of the test data is the normal test set, and it contains 9131 entries (30 minutes frequency).

After splitting, all data are decided to be scaled by the min-max scaler after trial and tested by fitting training data. This is done using the sci-kit-learn preprocessing package.

5.2 Algorithm Implementation

In the next chapters, the hyper-parameter settings and structures of the networks are discussed. For both Auto-encoder and VAE, the batch size is set to 256, and the number of epochs is set to 50. These are the common values found in similar works online.

5.2.1 One-class SVM

OCSVM(One-class SVM) works as a baseline model. Before fitting the data into OCSVM, PCA (principal component analysis) is used to extract important features. Components with a total explained variance larger than 95% are selected to proceed into the model. The OCSVM uses the 'RBF' kernel and the ν is set to be 0.394 after grid-search. The gamma, which is the kernel coefficient, is set to 'auto, which is equal to $\frac{1}{\text{number of features}}$.

5.2.2 Local Outlier Factor

The k (number of neighbors) in LOF is set to 72 by grid search.

5.2.3 Auto-encoder

Figure 13 shows the structure of the implemented Auto-encoder. The parameters are determined based on trial runs, The network starts with a 25 dimension input layer. In the encoder, the inputs first go through a drop-out layer, where input units are randomly set to 0 with a frequency of 0.5 to avoid over-fitting. After going through 3 dense layers, the dimension of the data is reduced to 3 to learn its latent representation.

In the decoder, the found latent dimensions are reconstructed by going through 3 hidden layers, where the activation function of the last layer is linear and the rests are 'relu'.

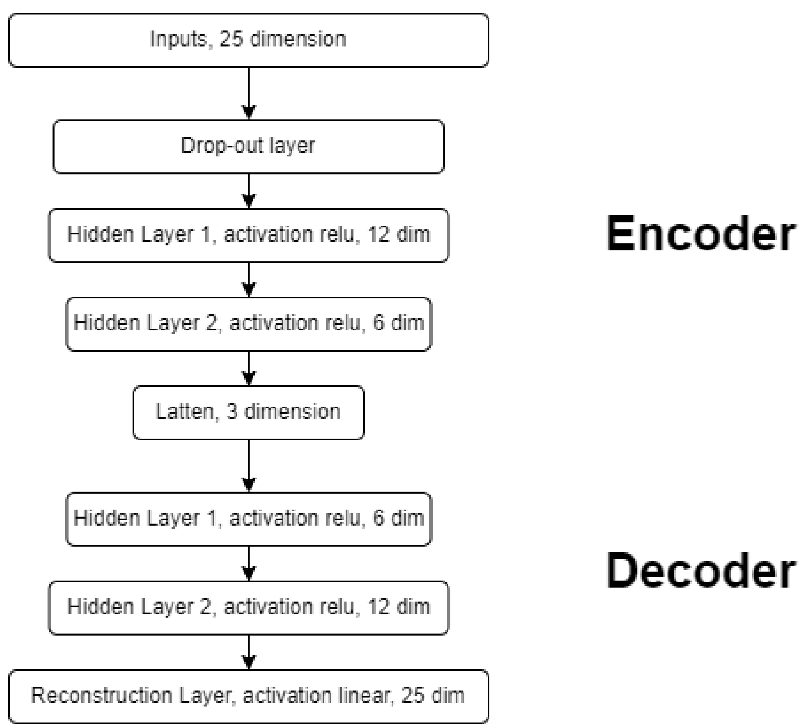


Figure 13: Structure Autoencoder

5.2.4 Variational Autoencoder

The VAE(Variational Autoencoder) follows a similar structure as the autoencoder. The parameters are determined based on trial runs, The inputs first go through a drop-out layer and then reduce to 5 dimensions after to hidden layers. The means and standard deviation vectors are then calculated to be sampled. The samples then go through 2 hidden layers in the decoder to reconstruct to data. (Some model performance results of VAE can be found in Appendix C)

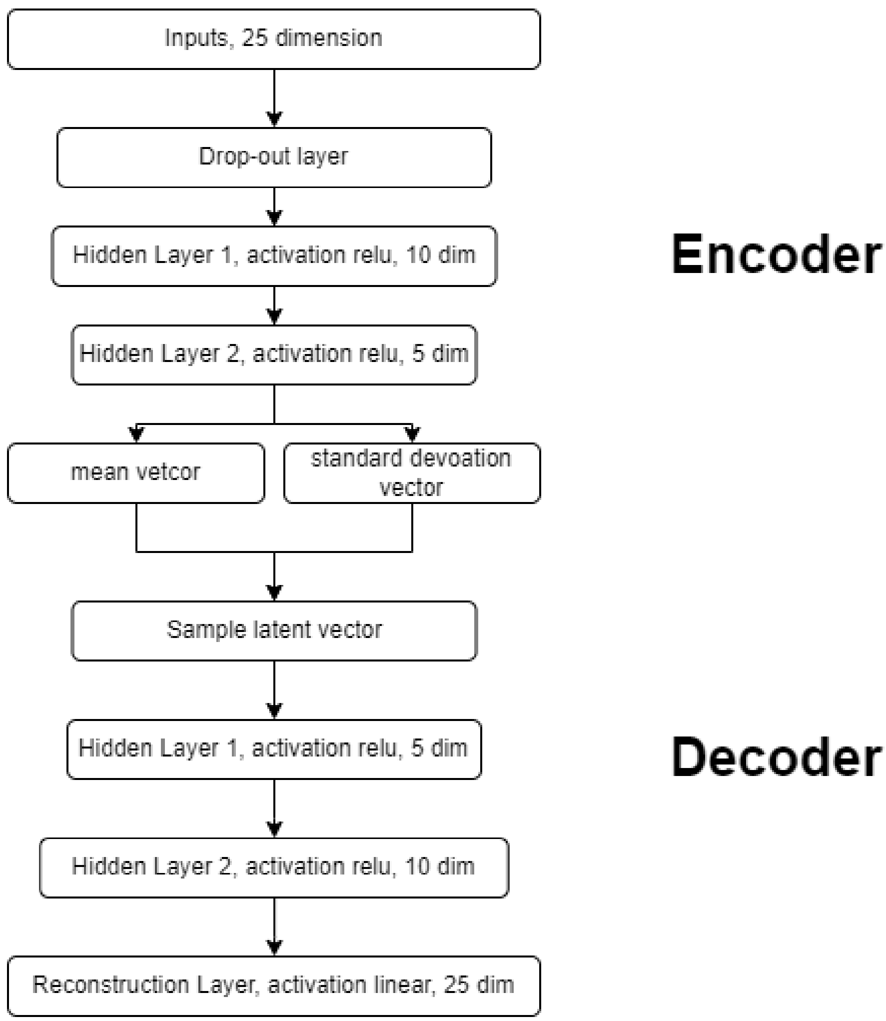


Figure 14: Structure VAE

5.3 Evaluation

In order to assess the results, two types of evaluation are employed: quantitative evaluation and qualitative evaluation. The quantitative evaluation examines measures such as accuracy, precision, recall, and F1-score, while the qualitative evaluation compares the ability of the predictions to successfully identify anomalies in the alarm code and incident report in advance. It should be noted that the damages recorded in the incident report are often approximations and the goal of anomaly detection is to predict abnormal events in advance. Therefore, a certain number of false positive predictions (i.e., predicting normal data points to be abnormal) is acceptable during the evaluation process.

6 Results

This chapter will focus on the results of Bridle 6 rather than Bridle 5 (which will be put into the appendix) because the alarm code (label) is only available for Bridle 6 hence the quantitative analysis can only be done on the results of Bridle 6. With the presenting results, the report will first present the findings about the best data preprocess method to build an anomaly detection model. Secondly, the report will discuss and evaluate the results of four different machine-learning algorithms.

6.1 Best Data Preparation Method

Datasets created by different re-sampling methods and time frequencies are experimented with to find the best-performance combination. Table 8 shows the results of them under the model OCSVM. As can be seen in Table 8, resampling to 3 minutes gives the worse result compared to the others. This could be because there is too much noise in the 3-minute data set compared to the others, as down-sampling reduces noise in data. It also has the longest running time due to its large data size. Hence we choose not to try the other resampling methods with this frequency.
The experiment suggested that resampling the data to 30 minutes frequency by taking the maximum will give the best prediction performance. Resampling to 60 minutes performs slightly worse. This may be due to its smaller data size. Among all resampling methods, taking the maximum gives the best result. This is probably because the purpose of the model is to detect anomalies, where taking mean and minimum may even out the abnormal data.

Resample methods	Resample time frequencies	F1-score	precision	recall	accuracy
Mean	3 minutes	0.664	0.99	0.497	0.502
Min	30 minutes	0.793	0.99	0.657	0.66
Max		0.865	0.99	0.763	0.765
Mean		0.846	0.99	0.734	0.736
Min	1 hour	0.794	0.99	0.659	0.663
Max		0.835	0.99	0.717	0.719
Mean		0.833	0.99	0.712	0.717

Table 8: Performance of different preprocessed data using OCSVM

Table 9 shows the differences between the predictions when using sub-bridles 6.1 and 6.2 and combining them as a complete dataset for Bridle 6. It can be observed that combining bridle 6 performs better than using data from just bridle 6.2, and it shows a similar performance to using bridle 6.1.

Figure 15 shows the differences in predicting anomalies in the incident report and alarm/fault codes. Each small gird in the figure represents a day. If the grid is red, the model predicts anomalies on that day.

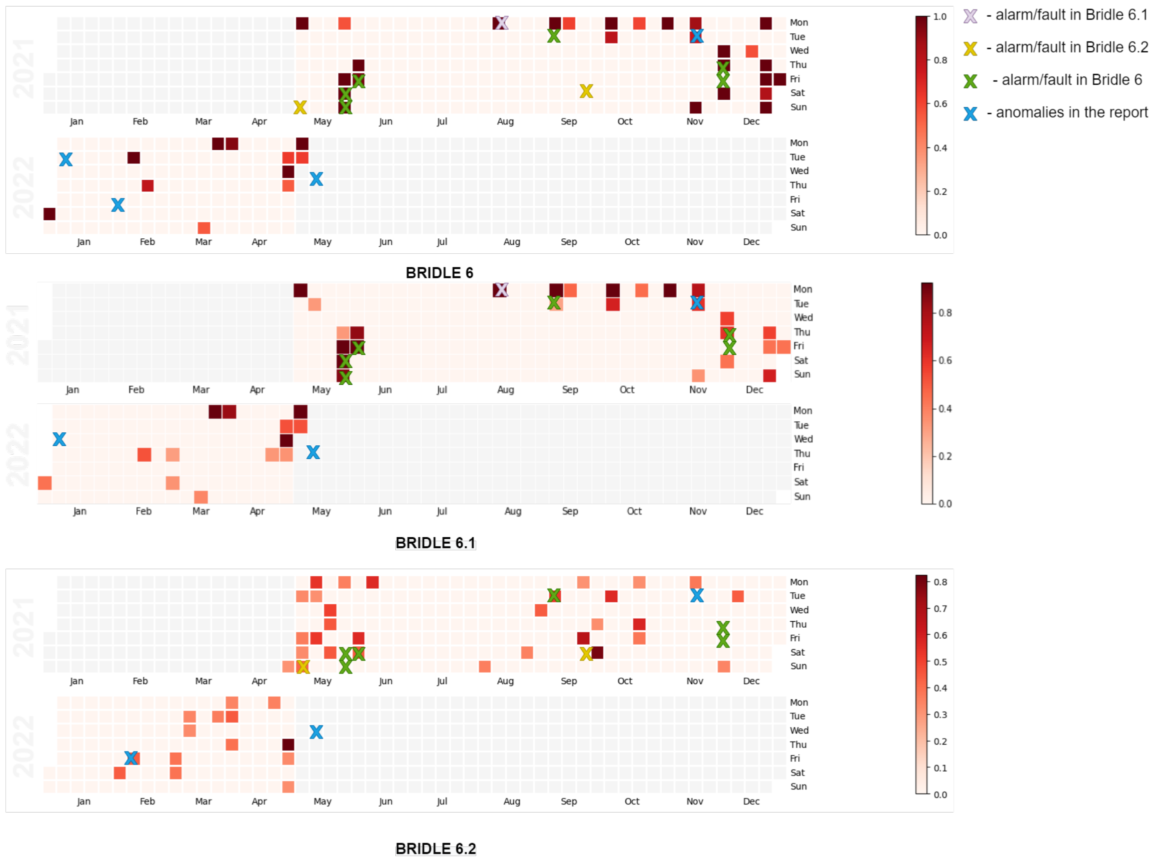


Figure 15: One-class SVM predictions for Bridle 6, Bridle 6.1, and Bridle 6.2 v.s. anomaly dates. The Red rectangle shows the model prediction, and the crosses show the anomaly dates on the fault/alarm code and incident report. Different colors of the crosses represent the difference in the anomaly record.

The shade of the red color represents the percentages of the number of anomalies over the total number of records on that day. The darker the red, the model predicts more anomalies on that day. The result is filtered by 50%, in other words, if more than half of the records are predicted as anomalies, it would be shown in the figure. This is done to filter out the days with only a few anomaly predictions, which are unlikely to be actual anomalies. The crosses represent the dates that either have an alarm/fault code or are recorded in the incident report (specific dates and bridle can be found in table 6 and table 7. The colors of the crosses show the differences. As can be observed in the figure, the model succeeds in finding 5 out of 13 anomalies on time and in advance, 2 anomalies on time, and 2 anomalies one day in advance. If the model is used for future anomaly detection, the predicted day of the anomaly may indicate the need for inspection or examination to find the cause. The model shows better prediction results in Bridle 6 and Bridle 6.1 than in Bridle 6.2. The predictions of combined Bridle 6 are similar to Bridle 6.1 but have fewer false positives.

The close relationship between the two sub-bridles in the production line may be the reason for the good performance in the combined dataset Bridle 6, even though the data is collected separately from two sub-bridles.

	F1-score	precision	recall	accuracy
Bridle 6	0.891	0.99	0.805	0.806
Bridle 6.1	0.891	0.99	0.804	0.806
Bridle 6.2	0.865	0.99	0.763	0.765

Table 9: Performance differences of sub-bridles and combine bridle using OCSVM

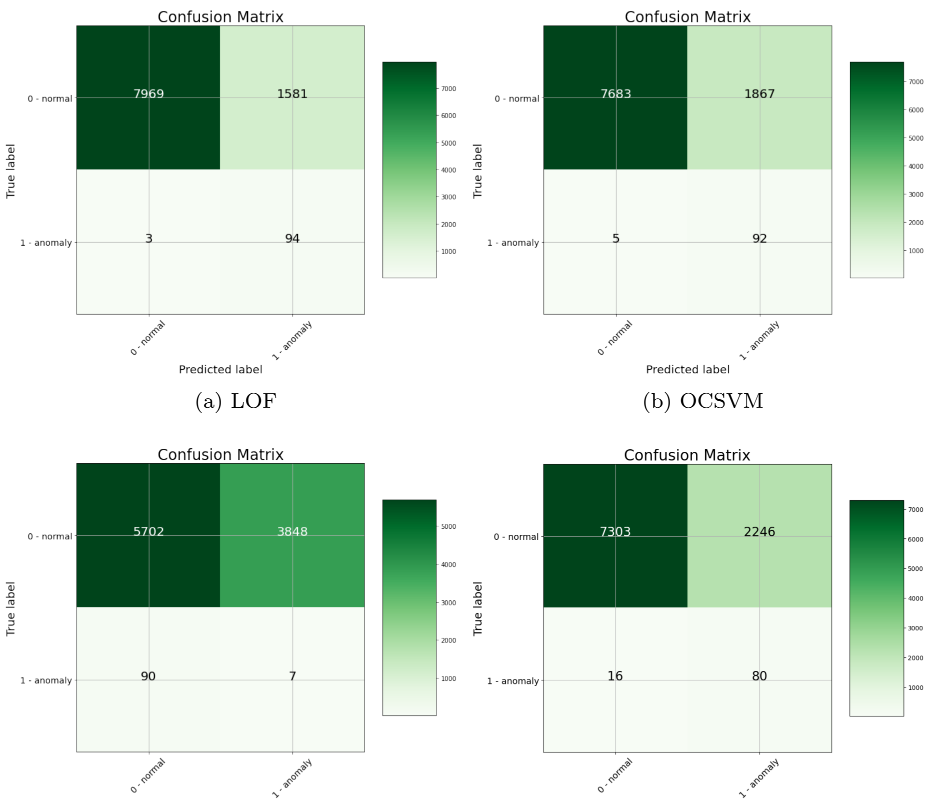
6.2 Model Performance

6.2.1 Quantitative Analysis

The data used in this chapter is prepared using the best-performance preparation methods found using the base-line model OCSVM in chapter 6.1 (i.e. The data is the combination of Bridle 6.1 and 6.2, resampling to 30 minutes frequency by taking the maximum in this frequency). For the other algorithms, this has been proved to be a least not worse than using a single sub-bridle by running some trial runs. As one can see in Table 10, Autoencoder shows the worse performance probably due to over-fitting, considering its low training loss. The model over-fits the data, hence even if the data deviates slightly from the average behavior, the autoencoder would indicate it as an outlier. LOF is the best performance model, and OCSVM and VAE have slightly lower performance. However, as discussed in chapter 3.4, the labels can only be used as an approximation of the anomalies. Hence performance will take the qualitative analysis into account. Figure 16 shows the specific confusion matrices of each model.

	F1-score	recall	precision	accuracy
LOF	0.912	0.834	0.99	0.838
OCSVM	0.893	0.831	0.99	0.814
Autoencoder	0.004	0.071	0.002	0.591
VAE	0.754	0.762	0.99	0.769

Table 10: The data used here is the combination of bridle 6.1 and 6.2, resampling by taking the maximum in 30 minutes.



(a) LOF (b) OCSVM

(c) Autoencoder (d) VAE

Figure 16: Confusion matrix of (a) LOF (b)OCSVM (c)Autoencoder (d) VAE.

6.2.2 Qualitative Analysis

Besides the quantitative analysis, it is significant to observe whether the model succeeds in finding the anomalies in the incident report and whether it predicts in advance. The visualization of such analysis is shown in figure 17. Again, a filter of 50% is applied.

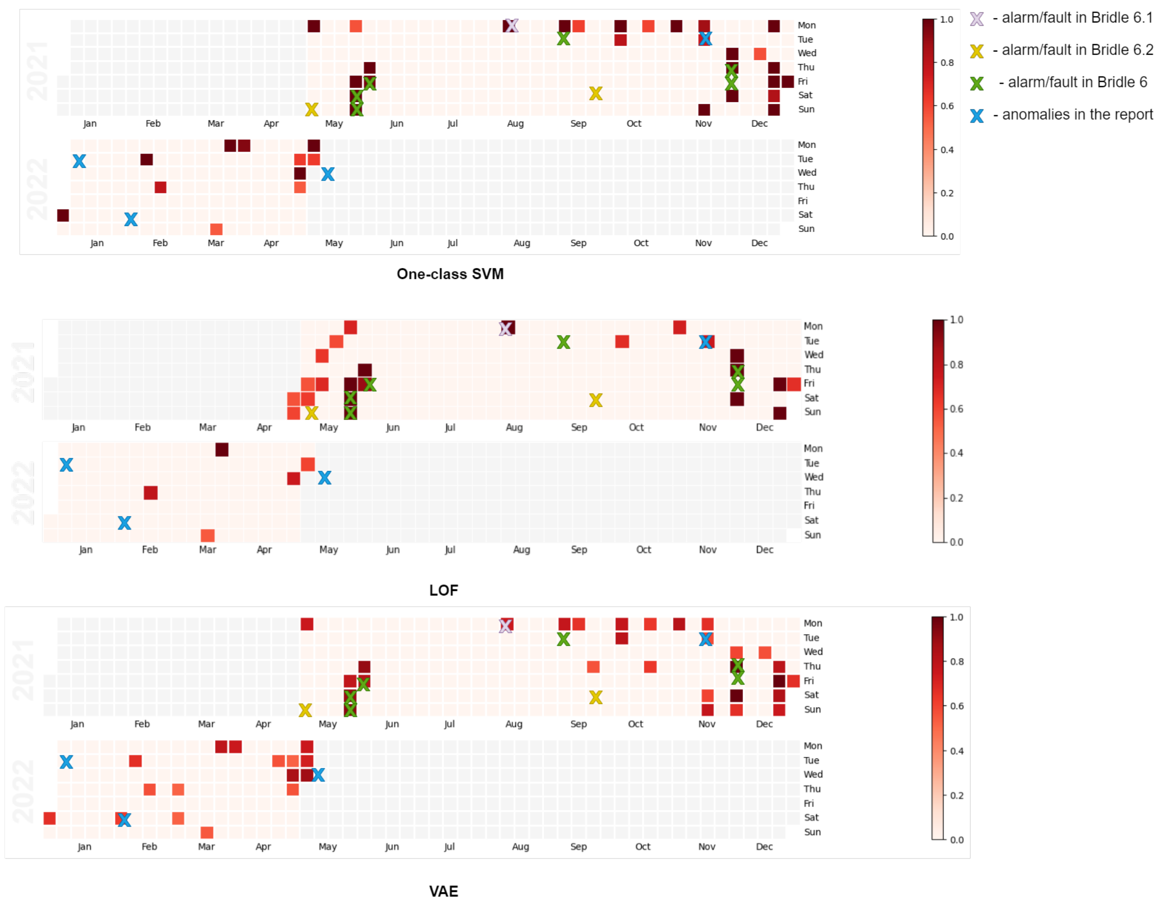


Figure 17: Predictions for Bridle 6 using OCSVM, LOF, and VA v.s. anomaly dates. The Red rectangle shows the model prediction, and the crosses show the anomaly dates on the fault/alarm code and incident report. Different colors of the crosses represent the difference in the anomaly record. If the blue cross only appears in the figure in one of the sub-bridles, it means that the incident report specifically mentioned the sub-bridle instead of the whole bridle.

LOF is the best-performance model when doing quantitative analysis, however, it fails to predict the alarm code in September 3rd 2021 while the others succeeded in predicting it one day in advance. VAE is the only model that succeeds in finding the anomaly on February 4^{th} 2022. Although this is just one anomaly, it is still useful since this is the damage that causes machine downtime, which stops production, as recorded in the incident report. Furthermore, it is an actual downtime that is not reported by the current maintenance system(i.e. the IBA fault/alarm code), hence it can be considered significant.

All models fail to find the two alarms/fault codes that only happened in Bridle 6.2, shown by the yellow cross, This is acceptable because the codes are just approximations. These two alarms only appear 3 and 4 times during the day respectively in only Bridle 6.2(as shown in table 7, hence it could be a false alarm. The last anomaly is on May 12^{th} 2022, which is the nearest actual breakdown. However, since the combined dataset IBA_SVM ends on May 2^{nd} 2022, the model can only predict its sign at the beginning of May. It is hard to say whether it is a good prediction or a false negative.
There could be several reasons why VAE and Autoencoder perform worse than LOF and OCSVM:

- VAE and Autoencoder have more complex structures than LOF and OCSVM. This report may not find the optimal hyper-parameters settings and structures yet.
- The labels used to calculate the F1-score and do the quantitative analysis are from the IBA alarm/fault code. They are the approximation of anomalies based on threshold setting. Linear-based algorithms LOF and OCSVM may perform better in finding outliers above the threshold.

6.3 IBA and SVM Investigation

Another aim of this report is to investigate the individual value of each data source. To do this, data from IBA and SAM4 are fitted into OCSVM respectively. Table 12 shows the performance of OCSVM. Comparing table 12 with the result shown in table 10, one can see that the combination of the two datasets performs slightly better. Each data source also gives reasonably good results on its own.

	F1-score	precision	recall	accuracy
IBA	0.886	0.99	0.795	0.797
SAM4	0.829	0.99	0.709	0.711

Table 11: Performance of individual data source on Bridle 6 using OCSVM

7 Conclusion

The goal of this study was three-fold. The primary objective was to investigate if anomaly detection models can be developed using Techport data. During this process, we assessed the presence of data-related challenges in Techport data and evaluated their impact on the accuracy of anomaly detection results. Additionally, we examined the potential benefits of combining data from multiple sources for anomaly detection and provided an analysis of the possible reasons for any lack of added value.

In this report, four different machine learning models (LOF, OCSVM, Autoencoder, VAE) are implemented and experimented with to predict anomalies using the Techport data. Apart from the worst performance model Autoencoder, the three models LOF, OCSVM, and VAE have their strengths and weaknesses. OCSVM and LOF have better F1 scores and higher recall than VAE. However, VAE succeeds in predicting some anomalies in the incidence report that actually cause downtimes and the others fail to predict, and it also has a reasonably good F1-score and the fastest processing speed. Furthermore, this damage was not detected by the original maintenance system (i.e. the IBA fault/alarm code). Therefore, it has the most potential among all models in a practical context. From this, we can conclude that anomaly detection models can be developed using Techport data.

The challenges of missing data, inaccurate and missing labels, mentioned in Challenges Data Quality under chapter 1 exist in this project and have a significant impact.

- The UTW data is not used and hence has no contribution to the final result due to its severe data missing rate. The most serious problem in the UTW data is its long periods of missing data, which makes it very hard to interpolate. Hence it cannot be fitted into the model on its own nor with the other data sources.
- Incorrect and inaccurate labeling is the most challenging part of this project. Techport cannot provide the exact timing of when the anomalies happened. Consequently, supervised learning algorithms cannot be used in this project. Moreover, inaccurate labeling also causes troubles when evaluating each model. The report can only give an approximate evaluation of each machine-learning model.

The report also investigates different preprocessing ways on the data and concludes that resampling the data into 30 minutes frequency by taking the maximum can give the best result using OCSVM. Also, it is found that modeling on bridle 6.1 outperforms that on bridle 6.2. And the model performance on data of bridle 6.2 is very similar to the combined data of bridle 6. This could mean that features from bridle 6.1 have more importance hence dominating the model result.

Each of the data sources is evaluated in this report to answer the last research question. UTW data cannot contribute to the model because of its long continuous missing data. IBA and SAM4 provide many features that can be used in the model. They can give reasonably well predictions by themselves. IBA performs almost as well as using the combination of them, but the combination slightly improves the model performance.

8 Suggestions and Discussion

One open challenge in the field of big data in predictive maintenance is data acquisition, particularly in regard to obtaining quality data. This collected data is often incomplete, poorly structured, or unannotated[8], which can present challenges for data science and machine learning tasks. Correct labeling is critical in these fields[9], and there are two main problems that can arise with labeling in the context of anomaly detection: missing labels and incorrect labels. Missing labels is when some data are not labeled, meaning that the indication of whether the data represents an anomaly or normal behavior is only available for a portion of the data[15]. Incorrect labeling occurs when anomaly data is labeled as normal or vice versa.

To improve anomaly detection, one of the most important steps is to improve the data quality. As discussed in Challenges Data Quality in chapter 1 and chapter 7, the UTW data cannot be used due to its poor data quality. UTW may record very valuable data but it cannot be used in machine learning due to its poor data quality. Reducing the missing data percentages and regularizing the data collection frequency to a constant time frequency may result in great progress in the Techport project.

Another important aspect is labeling. It is very significant to record the dates when anomalies took place as accurately as possible. This would enable the possibility to use more machine learning algorithms and highly improve the quality of evaluation of the model performance.

Practically, the company can make a daily or weekly monitoring system that checks the sensor data. This also can improve labeling if a failure is earlier observed. Moreover, adding extra monitors or systems that record and classify the failures and abnormal behaviors can also improve the label quality and enable more possibilities in predictive maintenance. Thirdly, the company can monitor the sensor running situation so that if sensors stop or are turned off, the workers can be informed. This can reduce the percentage of data missing. Lastly, sensors can be set to have to the same regular collecting frequency, which would make it easy to align and clean the data.

As mentioned in chapter 4.1, this report did not continue interpolating the missing values in the data but choose to drop all missing values. This can be improved in future works.

Appendix

A Data Overview: Basic information

The following table presents the basic information about the rest of the bridles from each data source. The data source will only include UTW and SAM4, as IBA produces the data in the same way.

Bridle 5.1				
	time span	Entries	Frequency	Feaures
SAM4	2019-12-31 to 2022-05-02	1639163	irregular	13
UTW	2020-09-30 to 2022-05-02	65085	irregular	54
Bridle 5.2				
	time span	Entries	Frequency	Feaures
SAM4	2019-12-31 to 2022-05-02	1783686	irregular	13
UTW	2020-09-30 to 2022-05-02	65005	irregular	51
Bridle 6.1				
	time span	Entries	Frequency	Feaures
SAM4	2019-12-31 to 2022-05-02	1540803	irregular	13
UTW	2020-09-30 to 2022-05-02	81170	irregular	58

Table 12

B Results Bridle 5

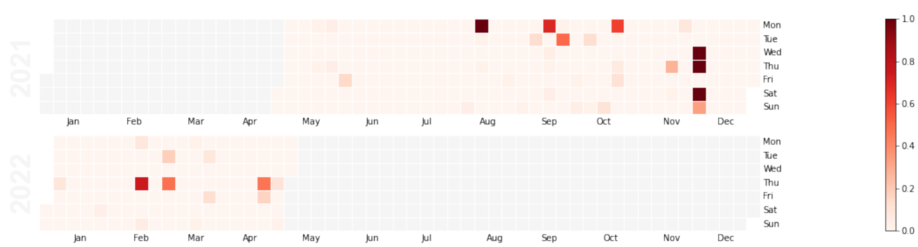


Figure 18: Prediction visualization Bridle 5 LOF

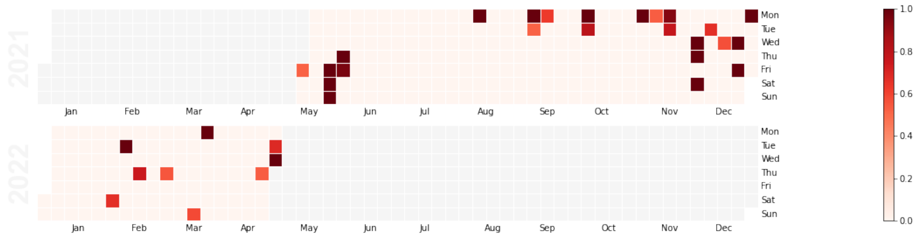


Figure 19: Prediction visualization Bridle 5 OCSVM

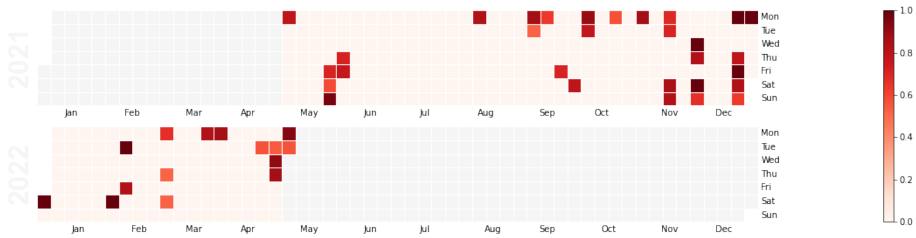


Figure 20: Prediction visualization Bridle 5 VAE

C Bridle 6 VAE Model Results

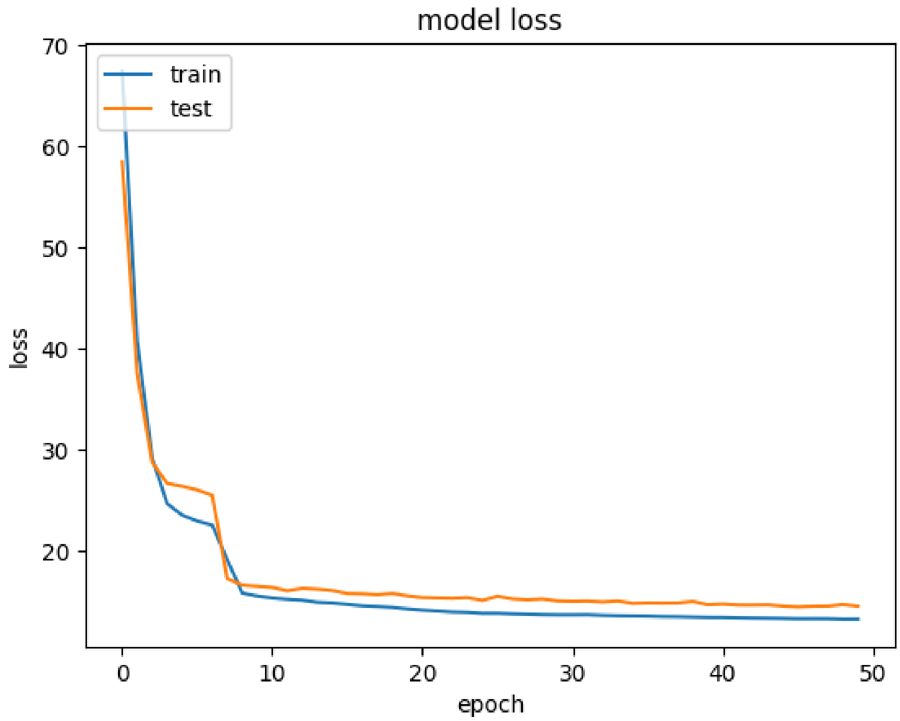


Figure 21: VAE Model Training Loss

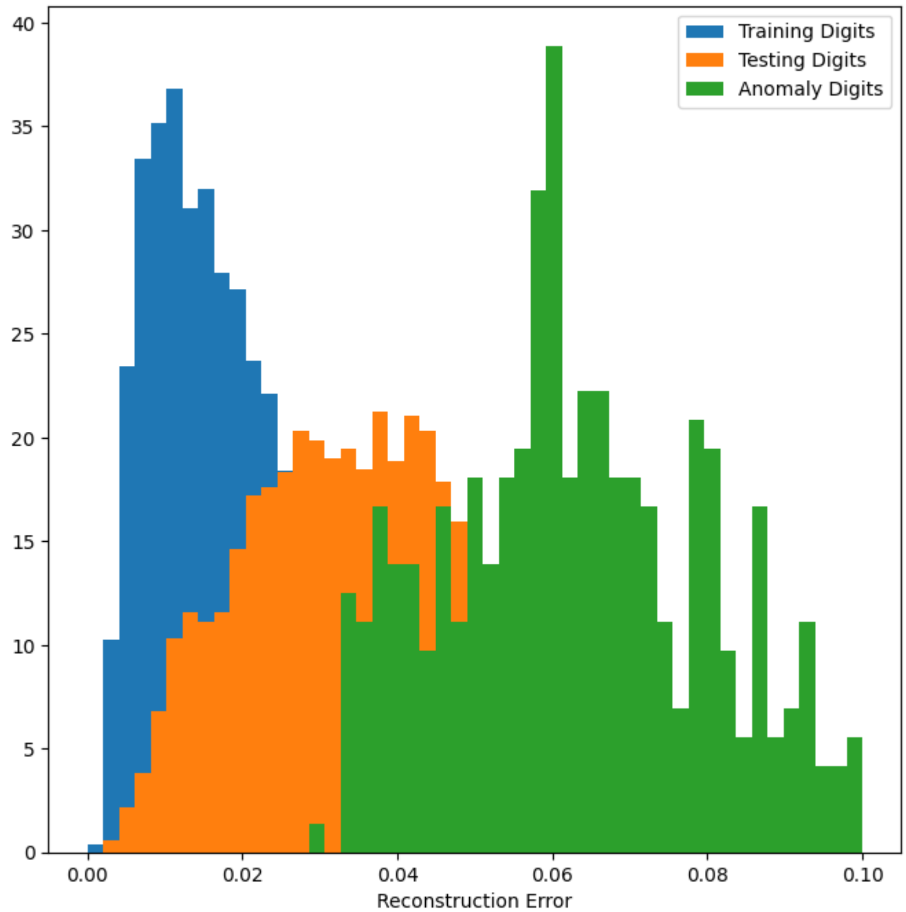


Figure 22: VAE Reconstruction Error Bridle 6

References

[1] Building a data science pipeline. https://www.youtube.com/watch?v=cugl5t-W1sE. Accessed: 2022-10-21.

[2] One-class svm with non-linear kernel (rbf). https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto$_{e}xamples/svm/plot_{o}neclass.html.Accessed$ $2022-10-04$.

[3] Pathways alignment with national / regional initiatives - first iteration. https://www.connectedfactories.eu/sites/default/files/d2.3$_{p}athways_{a}lignment_{f}irst_{i}teration.pdf.Access$ $2022-12-15$.

[4] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[6] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

[7] P. K. Chan, M. V. Mahoney, and M. H. Arshad. A machine learning approach to anomaly detection. Technical report, 2003.

[8] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, and J. Barbosa. Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, 123:103298, 2020.

[9] D. Hao, L. Zhang, J. Sumkin, A. Mohamed, and S. Wu. Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance. *IEEE journal of biomedical and health informatics*, 24(9):2701–2710, 2020.

[10] P. Kamat and R. Sugandhi. Anomaly detection for predictive maintenance in industry 4.0-a survey. In *E3S Web of Conferences*, volume 170, page 02007. EDP Sciences, 2020.

[11] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[12] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[13] T. Lane and C. E. Brodley. An application of machine learning to anomaly detection. In *Proceedings of the 20th national information systems security conference*, volume 377, pages 366–380. Baltimore, USA, 1997.

[14] W.-C. Lin, C.-F. Tsai, and J. R. Zhong. Deep learning for missing value imputation of continuous data and the effect of data discretization. *Knowledge-Based Systems*, 239:108079, 2022.

[15] Y. Liu, K. Wen, Q. Gao, X. Gao, and F. Nie. Svm based multi-label learning with missing labels for image annotation. *Pattern Recognition*, 78:307–317, 2018.

[16] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. In *Handbook of neural network signal processing: Neural network signal processing*, pages 95–134. CRC Press, 2002.

[17] S. Nijman, A. Leeuwenberg, I. Beekers, I. Verkouter, J. Jacobs, M. Bots, F. Asselbergs, K. Moons, and T. Debray. Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review. *Journal of Clinical Epidemiology*, 142:218–229, 2022.

[18] S. Omar, A. Ngadi, and H. H. Jebur. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, 79(2), 2013.

[19] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.

[20] A. J. Smola and B. Schölkopf. Learning with kernels: Support vector machines, regularization, optimization, and beyond, 2002.

[21] B. Sun, S. Zeng, R. Kang, and M. G. Pecht. Benefits and challenges of system prognostics. *IEEE Transactions on reliability*, 61(2):323–335, 2012.

[22] D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo. *Advances in neural information processing systems 8: Proceedings of the 1995 conference*, volume 8. Mit Press, 1996.

[23] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.