Exploring Anomaly Detection Methods Using Artificial Intelligence

Morgan Huang

2nd year Statistics

September 2023 - December 2023

1st co-op term

Table of Contents:

Executive Summary

In this report, I will be presenting my duties for the past 4 months at Hatch Ltd., and I will be presenting a paper that I wrote on my research on anomaly detection, a topic that I have researched for the last month of my co-op work term.

Overview of Work Term

For the past 4 months, I have been working at Hatch Ltd., an international engineering consultant firm based in Mississauga, Ontario. My role as an Analytics and Decision Optimization Co-op Student is to work with the Digital Team in assisting them with creating digital products and solutions, solving problems in large-scale environments using primarily digital tools.

As an engineering firm, Hatch primarily specializes in the mining industry, with around 70 percent of its resources used to solve problems in the mining area. While the digital team focuses on many different industries such as civil and industrial industries, I was tasked with working on a tailings project, a project based in the mining industry, and I was tasked with building a digital model to help predict the influent flow of the tailings pond. In short, tailings are a by-product of processed ore: chemicals are used to separate minerals from rocks. Once the minerals are separated, the left-overs are the tailings, which contain harmful chemicals that must be stored in a tailings pond to avoid contamination with neighboring environments. Predicting the flow of the tailings pond with respect to time and weather data is crucial to prevent the tailings from overflowing the pond and protect ecosystems and nearby cities.

However, due to problems collecting pond data, I was only able to work with 80 percent of the actual data, and thus I was not able to make further progress other than exploratory analysis and training models; I was not able to be involved in the actual implementation of the model. Therefore, I was then placed to do research and development on the topic of anomaly detection to help assist with other projects that my colleagues were working on for the rest of the work term.

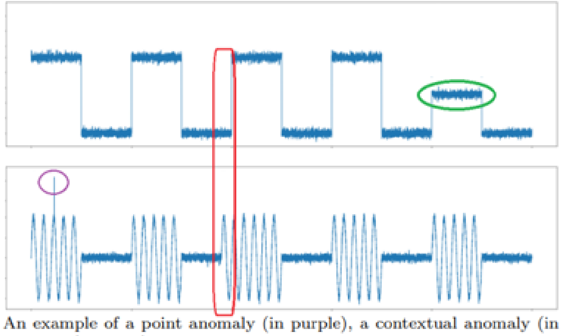
1. Introduction

In many industries such as in finance, software, and in engineering, companies rely on collecting data and performing data analysis to create future business products and solutions. Whether it is to predict future stock market prices to identify future investments, or outlining the trend and seasonality in revenue gained by a business to understand their financial strengths, data analysis has been a key aspect in providing useful quantitative information for businesses. Data analysis is also key in detecting faulty information in a product, such as a broken sensor, or faulty technology. This field of data analysis is what we call Anomaly Detection, and is heavily used in engineering firms to analyze potential faults in sensors of an engineering process, such as a chemical plant or a manufacturing plant.

The field of anomaly detection, while useful, is a relatively new field in math and statistics, and has undergone many new techniques in finding anomalies in data. When the field of data analysis came into businesses, companies would use basic statistical techniques to analyze data and find anomalous data. For instance, we can derive probability distributions on data, and if we detect that the distribution has two peaks in the data, also called a bimodal distribution, we can assume that an outside force has been placed on the data, thus it is reasonable for the analyst to assume that a fault has occurred when collecting data. Roughly into the new millennium, machine learning became a new innovation, and is able to learn linear and non-linear relationships between features in data, relationships that the naked eye is not able to detect. Recently, deep learning methods, a subset of machine learning that relies on neural networks, have been implemented in anomaly detection. In this report, I will be showing you my research on different anomaly detection techniques, specifically in machine learning and deep learning, and their implementations in real world industries.

2. Definition of Anomaly

To start analyzing anomalous data, one must know what an anomaly is, and what kind of data he is analyzing. In reality, there is no concrete definition of what an anomaly is; we can only identify anomalies visually and intuitively. With respect to engineering processes, these processes collect data in the form of a time series: a series of data that is influenced by time. In the realm of time series data, there are 3 main types of anomalies: point anomalies, collective anomalies, and contextual anomalies. Briefly, a point anomaly is a data point that differs from the rest of data, a collective anomaly is a group of data points that are considered anomalous, however individual points may not be anomalous, and a contextual anomaly is a data point that is considered anomalous in one context, but not in another. We call a non-anomalous point a normal point.



(Audibert, 2022)

Figure 1: An example of a point anomaly (in purple), a contextual anomaly (in red) and a collective anomaly (in green)

3. Problem Definition

3. a) Supervised learning

Let us now formulate the problem we are trying to solve. In machine learning, there are two main techniques: supervised learning and unsupervised learning.

In supervised learning, there are two components needed: an input and an output. A machine learning model trains by collecting data that contains the input and output, and the model trains by learning relationships between the input and output. Once the model is trained, the model contains clear mathematical instructions on how to transform input data to create its output. Therefore, after training, we can input into the model new data and the model, through its instructions, will create predictions for its output. In the context of anomaly detection, the model learns the relationship between the data(input) and whether or not each corresponding data point is anomalous or normal(output). The model is then expected to predict future data and whether or not these future data points are anomalous or not.

3. b) Unsupervised learning

In unsupervised learning, we only require the input data. Instead of the model learning the relationship between the input and the output, the machine learning algorithm learns the relationships inside the input data, either clustering the data into different groups based on certain attributes, or separating them into different components. After training, the model is then able to predict how to divide future data into different clusters or components.

Creating an unsupervised learning problem with anomaly detection is a little more complex than with supervised learning. Because we do not want the model to train on anomalous data, as this will result in the model compromising its clustering or component structures in order to fit the anomalies, we must split the dataset into two parts: a normal dataset and a faulty dataset. We then train the normal dataset to learn the patterns in the data, which gives clear instructions on how to separate them into different clusters. After training, when we input data into the model, it reconstructs the data in order to fit these different clusters, and thus we can calculate the reconstruction error, the difference between the data reconstruction and the data itself.

Because the model has been trained on normal data, we can expect the reconstruction error between the normal data reconstruction and the normal data to be low. We can also expect that the reconstruction error between the anomalous data reconstruction and the anomalous data to be high, as the model is not trained to know how to classify the anomalous data points into the created clusters. Therefore, we classify anomalous data points as data points that have a high reconstruction loss, and classify normal data points with low reconstruction loss.

There are different ways of defining quantitatively what a high reconstruction loss is. A simple way is to define it as the maximum reconstruction loss of a certain data point from the normal dataset. However, in many cases, some anomalies will not be detected, if the maximum loss is too high. Therefore, we can define a threshold value; any point above the threshold value is considered anomalous, and points below the threshold are considered normal. We can use statistical methods to calculate the most optimal threshold value: we test multiple different threshold values, then, we choose the threshold value that holds the highest F1 score. An F1 score is an accuracy metric, similar to accuracy, but also takes into consideration the falsely identified data points.

In many business cases, because it is hard to gain access to data that is fully labeled: every data point has a classification on whether or not the data point is anomalous or not, we usually approach this problem in an unsupervised way. We are now ready to explore different machine learning methods that are used in anomaly detection.

4. Methodology

4. a) PCA(Principal Component Analysis)

Principal Component Analysis is a machine learning algorithm that learns how to label data into different components, specifically into different dimensions. For instance, if we have data that is 10 dimensions, and we want to perform PCA for 6 dimensions, the model will then learn how to split the data into those 6 dimensions by sending data points into its ‘nearest optimal dimensions’, using a linear algebra technique called eigenvalue decomposition. These dimensions are chosen based on its variance of the data; a feature in the dataset with high variance will have its dimension be more likely to be chosen than a feature with low variance.

We can implement PCA into our anomaly detection problem. By choosing an optimal amount of components, the model is able to split the normal data into its most important features and relationships. Once we input anomalous data into PCA, we can expect that the model will not know how to split the data, thus leading to a high reconstruction loss.

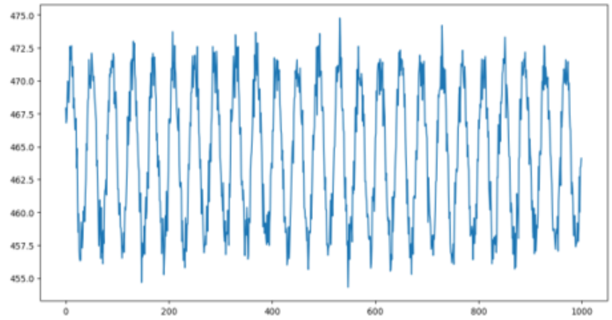
4. b) Autoencoder

Autoencoders are a standard feedforward neural network, except that its output is equal to its input. A feedforward neural network is a deep learning method that takes its input and describes the input in the form of neurons, each neuron given a numerical weight. Once we traverse through each layer of the neural network, we describe each deeper neuron as a linear combination of the neurons from the previous layer, until we get to the output layer, which transforms the neurons back to the form of the output. During training, the weights that are given to each neuron is random, thus it is the job of the model to learn what weights are optimal such that the given output is correctly predicted, through techniques called backpropagation and gradient descent.

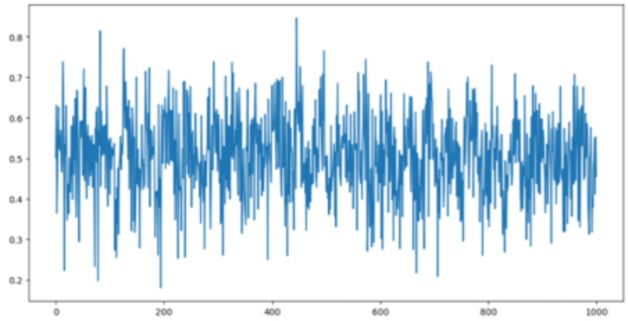
Because an autoencoder has its output as its input, instead of the neural network learning the relationships between its input and output, it learns the relationships inside the input data. Therefore, the model tries to recreate the input into a more simplistic manner, only identifying the most significant relationships in the data. When we input faulty data into the model, we can expect that it doesn't have those relationships; thus the model does not know how to reconstruct the faulty data, and therefore fails the reconstruction process.

5. Experiment

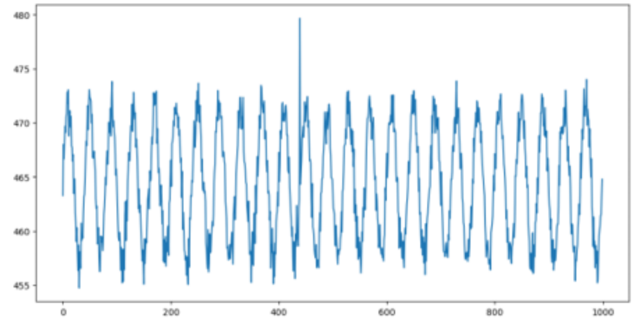
To test the anomaly detection problem, I have created some fake data that resembles engineering process data, and evaluated its results. I have created a dataset that contains 50 different features, each with 1000 timestamps, displaying a random function of sine or cosine. Each function of sine has random frequencies, amplitudes, mean value, and white noise. I then split the dataset into two; the first being a normal dataset, and the second being a faulty dataset.



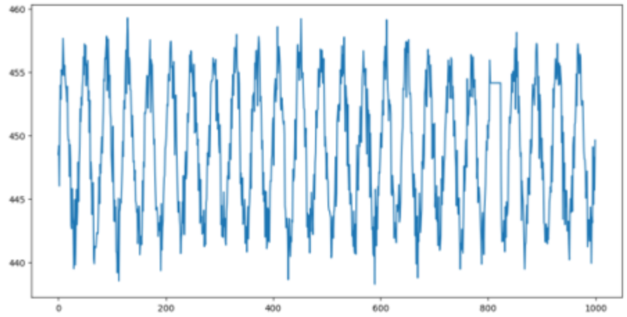
Normal Data: Feature 49



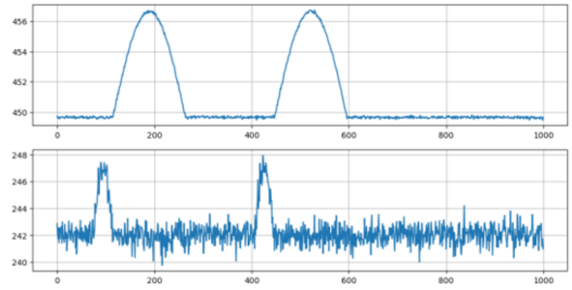
Normal Data: Feature 18



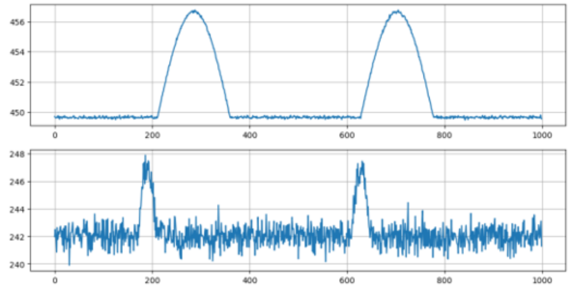
Faulty Data: Point Anomaly



Faulty Data: Collective Anomaly



Normal Data

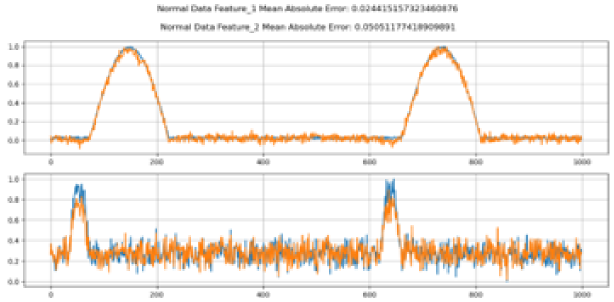


Faulty Data: Contextual Anomaly

To define the threshold value, I tested with 200 different threshold values from 0.001 to 0.2, then I would test using both the normal dataset and the faulty dataset each threshold value, and calculate its F1 score. The threshold value that is optimal is the one that corresponds to the maximum F1 score over all threshold values tested. In addition, the F1 score must be between values 0 and 1.

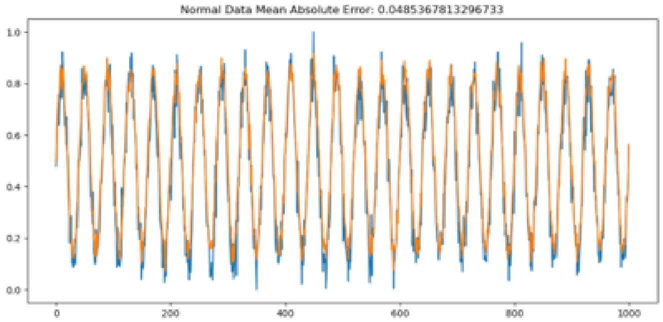
6. Challenges

There were a couple of challenges that have been presented as I performed this experiment. Firstly, when testing if the model detects contextual data, as shown from the figure below, the model is able to predict the 'peaks' of the data perfectly, despite the oscillations being completely random. This shows that PCA and Autoencoders do not take data sequentially, meaning that we would have better results if we used a prediction model instead, such as Long-Short Term Memory Autoencoders(LSTM-AE) and Auto-Regressive Integrated Moving Average(ARIMA), as these models take data sequentially, predicting future values based on past values. However, this implies that the features must not be random, and there must be a recurring pattern throughout the dataset in order to predict effectively.

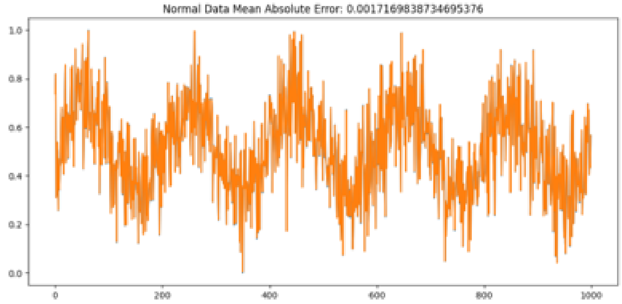


PCA Normal Data Reconstruction. Random peaks do not matter in model reconstruction

The second challenge that I encountered in this problem is dealing with high varieties of noise in different features in the dataset. With respect to PCA, if a feature has relatively low noise, then PCA would generalize well with the data. However, if a feature has a lot of noise, then with enough components, PCA does not care to generalize the feature at all, due to the high variance of the feature, and reconstructs the data almost perfectly. This causes a problem in detecting anomalies when the anomaly is situated in the noisy feature, as because PCA will not generalize, it is impossible for the anomaly to be detected by the model, regardless of the threshold value we set.

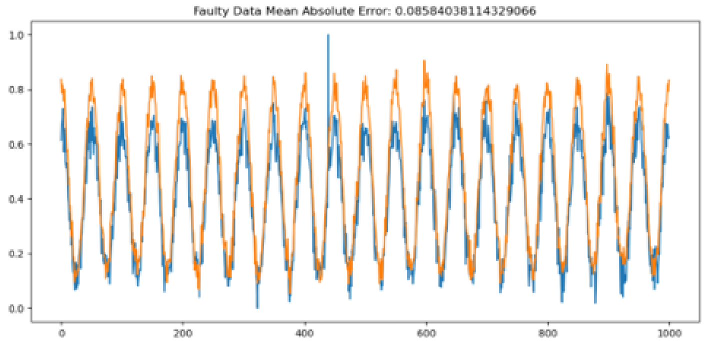


PCA on low noise feature

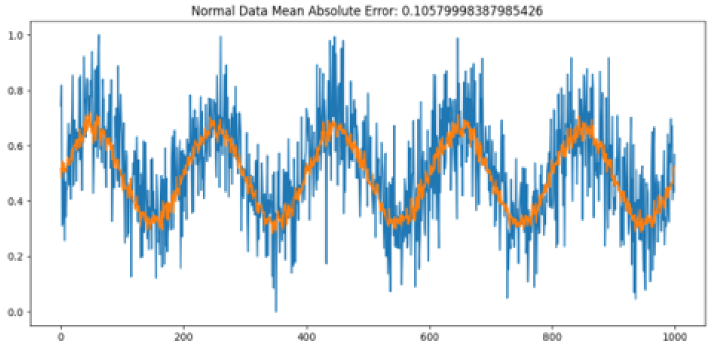


PCA on high noise feature

With respect to Autoencoders, the challenge arises in defining the threshold value for the model for noisy data. This is because unlike PCA, autoencoders do attempt to generalize noisy features. However, it follows that the error value between the reconstruction and the original data is so large that it becomes hard to define a threshold value that is able to compensate the large error values found in noisy features and the error value between an anomaly and its reconstruction in a feature with less noise, where the error values of the anomalies are typically less than the error values of the normal reconstruction of the features. Therefore, many anomalies are not able to be detected by the model due to the difficulty of finding a suitable threshold value.


Faulty Data Mean Absolute Error: 0.08584038114329066

Autoencoder on point anomaly with low noise feature


Normal Data Mean Absolute Error: 0.10579998387985426

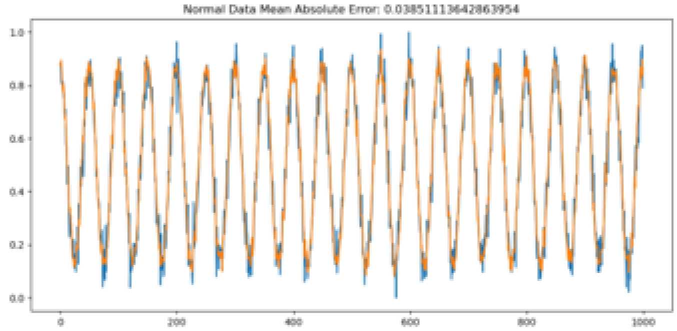
Autoencoder on high noise feature

Some solutions that can be used to solve these issues is to use denoising techniques to level out the high variety of noise found in different features, then training the model for anomaly detection. For PCA, this would mean that we are able to detect anomalies in noisy features, and for autoencoders, we would be able to define a suitable threshold value. Another solution to this problem is to define a threshold value for each feature. Thus, a feature with a lot of noise would have a larger threshold value then a feature with no noise, and is able to consolidate the differences in variance in the data. This, however, is at the expense of computational complexity; we must test different threshold values for each feature instead of different threshold values for the entire dataset.

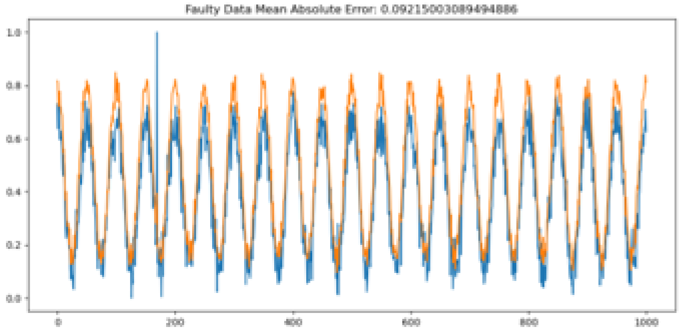
7. Results

For this paper, I have only examined the quantitative results of detecting point anomalies, leaving quantitative analysis on collective anomalies and contextual anomalies for future work.

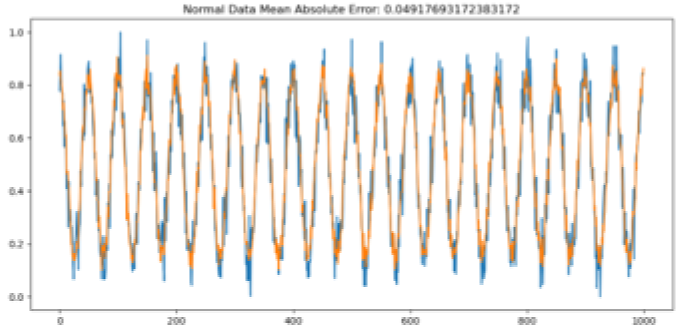
Because of the challenges presented in the previous section, the autoencoder fails to perform well on any of the threshold values, while PCA does a decent job in anomaly detection, with its optimal threshold value corresponding to an f1 score of over 0.75.



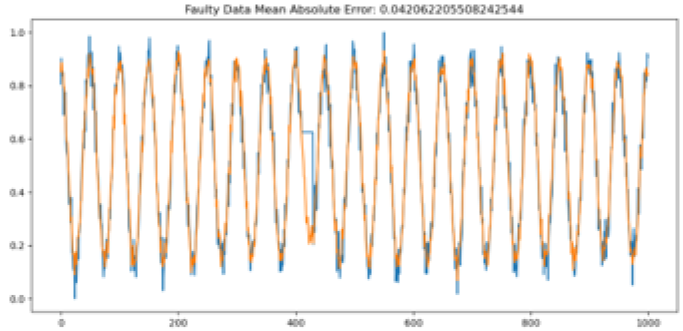
PCA: Normal Data



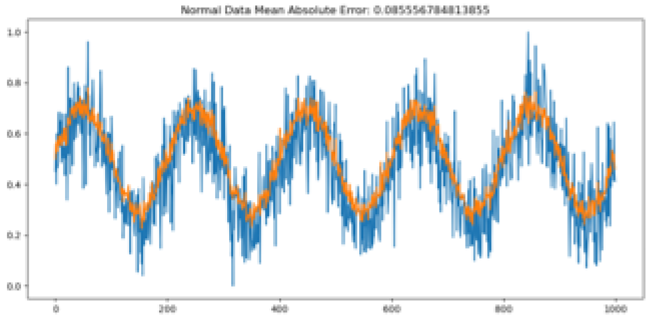
PCA: Point Anomaly



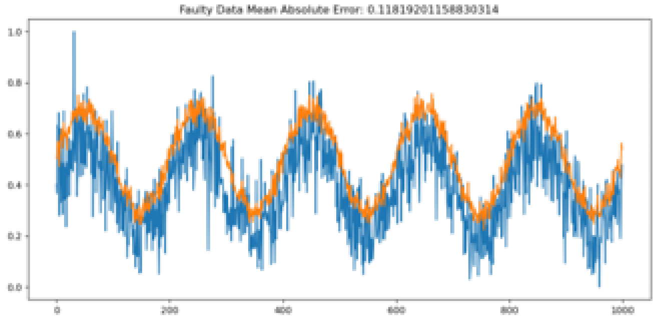
PCA: Normal Data



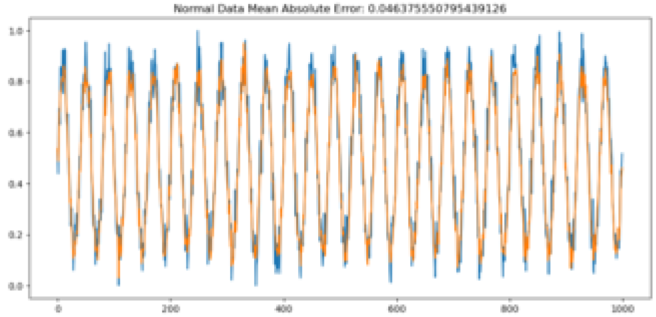
PCA: Collective Anomaly



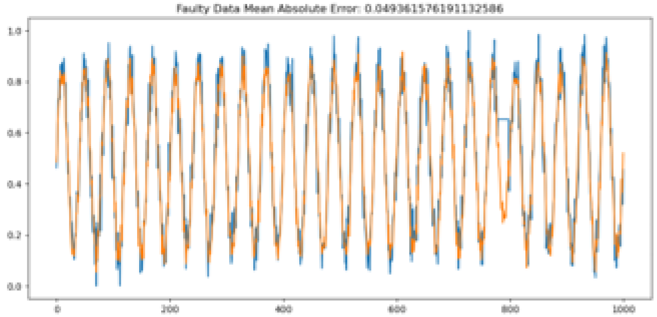
Autoencoder: Normal Data



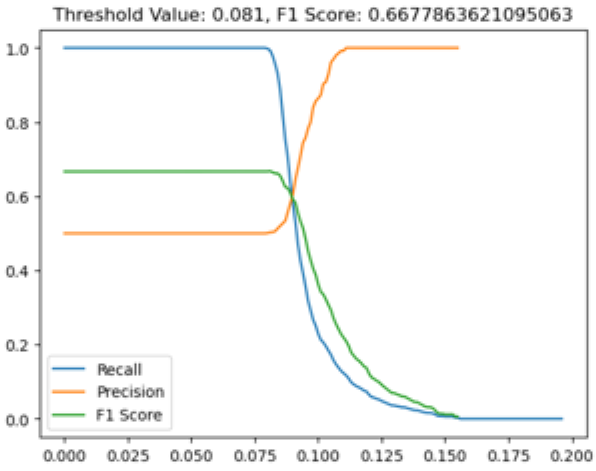
Autoencoder: Point Anomaly



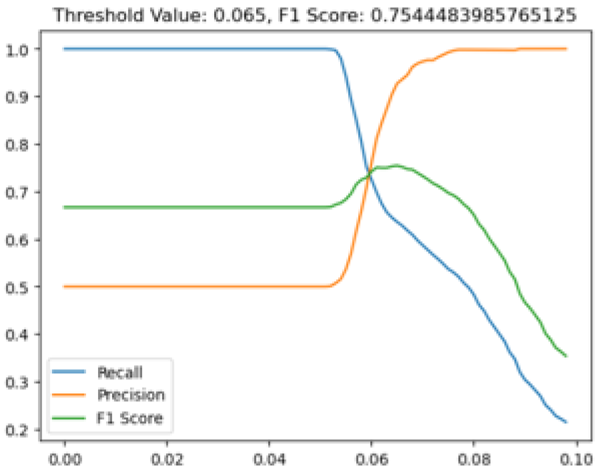
Autoencoder: Normal Data



Autoencoder: Collective Anomaly

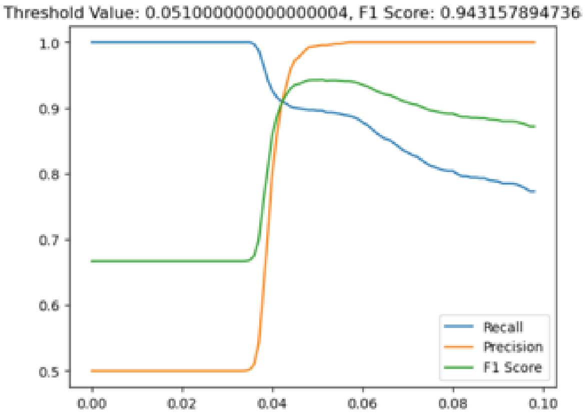


F1 Graph: PCA

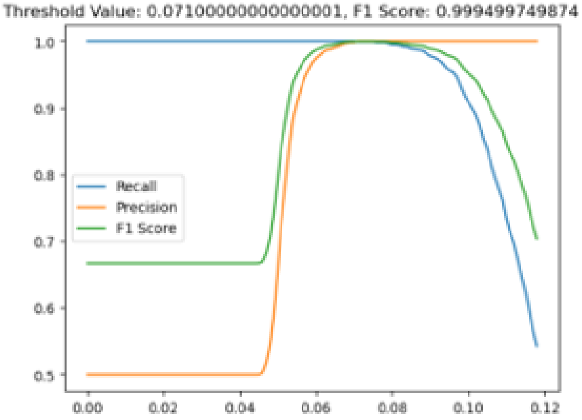


F1 Score: Autoencoder

However, when we lower the variety of noise in the dataset, we can see that the results are a lot better for both models, with autoencoders providing almost a perfect f1 score in detecting anomalies. Therefore, we can conclude that the variance of noise between features in a dataset is crucial in the problem of anomaly detection.



F1 Score: PCA on low noise data



F1 Score: Autoencoder on low noise data

8. Future Work

As mentioned above, I would like to experiment with detecting collective anomalies and contextual anomalies as well, which the latter would require an implementation of predictive models. In addition, I would also like to test with more complex functions, with sine waves changing frequencies with respect to time, and implement denoising techniques on normal and faulty data, as most engineering processes work with very noisy data.

9. Conclusion

Anomaly detection is an important field in many different industries, such as finance and engineering, due to its large usage of data collection and data mining. Anomaly detection methods have evolved to using statistical methods, to implementing machine learning models, and more recently, a focus on the usage of deep learning models.

In this paper, I have demonstrated possible solutions in anomaly detection, providing results from PCA(Principal Component Analysis) and Autoencoders, given an artificial dataset of 50 features, each corresponding to a different function with a variety of noise. I have demonstrated the model's effectiveness on detecting point anomalies, and have presented solutions on different challenges that I have faced during experimentation. For example, when encountering a variety of noisy data, I have suggested to either implement denoising techniques, or to define a different threshold value for each feature. Given the effectiveness of its detection on point anomalies, especially given datasets that have been denoised, it is not effective however in detecting contextual anomalies, as these models do not take data sequentially, thus prediction models would suit this detection problem.

In summary, PCA and Autoencoders are proven to be useful in detecting anomalous data, however this is only when encountered with simple sine functions; more experimentation is needed to detect anomalies in real-world datasets.

10. Bibliography

1. Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2022). Do deep neural networks contribute to multivariate time series anomaly detection? *Pattern Recognition*, *132*, 108945. https://doi.org/10.1016/j.patcog.2022.108945
2. Crépey, S., Lehdili, N., Madhar, N., & Thomas, M. (2022). Anomaly detection in financial time series by Principal Component Analysis and Neural Networks. *Algorithms*, *15*(10), 385. https://doi.org/10.3390/a15100385
3. Tziolas, T., Papageorgiou, K., Theodosiou, T., Papageorgiou, E., Mastos, T., & Papadopoulos, A. (2022). Autoencoders for anomaly detection in an industrial multivariate time series dataset. *ITISE 2022*. https://doi.org/10.3390/engproc2022018023