Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Patrick Bjørkhaug Johannessen
Magnus Westerheim Johannessen

MIURA: Memory-efficient and Incrementally learning Unsupervised Real-time Anomaly detection for time series data

Bachelor's thesis in Digital Infrastructure and Cyber Security
Supervisor: Jia-Chun Lin
Co-supervisor: Ameen Chilwan
May 2024

NTNU
Norwegian University of
Science and Technology

Patrick Bjørkhaug Johannessen
Magnus Westerheim Johannessen

MIURA: Memory-efficient and Incrementally learning Unsupervised Real-time Anomaly detection for time series data

Bachelor's thesis in Digital Infrastructure and Cyber Security
Supervisor: Jia-Chun Lin
Co-supervisor: Ameen Chilwan
May 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Abstract

Orange Business is a leading provider of critical IT solutions and infrastructure, aiming to provide extensive real-time monitoring and analysis of various system metrics. The company seeks a system capable of detecting abnormal behavior across different systems to enable rapid incident response. The main goal of this project was to extend the deep learning anomaly detection approach, RePAD2, with the concept of incremental learning and to investigate how it would affect detection accuracy and time efficiency. This led to the development of the MIURA approach. To support MIURA, a data streaming and visualization architecture was needed that would enable fast and reliable data transfer, along with the capability to visualize detected anomalies. To provide context to the proposed solution, a greater focus was placed on reviewing relevant literature, which led to deeper insights into the field of anomaly detection. We conducted a series of evaluations using several real-world open-source time series, where we evaluated what effect different scalers had on detection accuracy. The results show that the MIURA approach provides better detection accuracy and time efficiency compared to RePAD2 when a time series exhibit minor fluctuations. We hope that the work done in this project will benefit further research. Therefore, we have chosen to open source the code of MIURA, scripts used in evaluation, and other relevant material. The repository is made available on GitHub.

Sammendrag

Orange Business er en ledende leverandør av kritiske IT-løsninger og infrastruktur, og har som mål å tilby omfattende sanntidsovervåking og analyse av ulike systemmålinger. Selskapet ønsker et system som kan oppdage unormal atferd på tvers av ulike systemer for å kunne reagere raskt på hendelser. Hovedmålet med dette prosjektet var å utvide den dyplærningsbaserte løsningen for anomalideteksjon, RePAD2, med inkrementell læring for å undersøke hvordan det ville påvirke deteksjonsnøyaktigheten og tidseffektiviteten. Dette førte til utviklingen av MIURA. For å støtte MIURA var det behov for en datastrømmings- og visualiseringsarkitektur som muliggjorde rask og pålitelig dataoverføring, sammen med muligheten for å visualisere oppdagede anomalier. For å sette den foreslåtte løsningen i kontekst, ble det lagt vekt på å gjennomgå relevant litteratur, noe som førte til dypere innsikt i feltet anomalideteksjon. Vi har gjennomført evaluering ved bruk av en rekke tidsserie-datasett som ligger åpent på nett, hentet fra reelle datakilder, hvor vi evaluerte effekten ulike skaleringsmetoder har på deteksjonen. Resultatene viser at MIURA oppnår bedre deteksjonsnøyaktighet og tidseffektivitet sammenlignet med RePAD2 på dataset som viser mindre drastiske svingninger. Vi håper at det arbeidet som er blitt gjort i dette prosjektet vil bidra positivt til videre forskning. Av den grunn velger vi å åpne kildekoden til MIURA, scriptene brukt til evaluering, og annet relevant materiale. Hele kodebasen er tilgjengeliggjort på GitHub.

Preface

The initial topic for this bachelor project was given to us by Orange Business, and though the main focus of the task was tweaked, the technologies and ideas were a great inspiration to our solution. Through the guidance of our supervisors we have managed to not only produce a thesis we can be proud of but, a solution that we hope could benefit further research. We would like to express our utmost gratitude to our supervisors Jia-Chun Lin and Ameen Chilwan who have made this project possible. With a special thanks to Jia-Chun Lin who helped us tremendously in making both the project and this thesis is as good as it could possibly be. And finally, a big thank you to all friends and family members who proofread this thesis, as we could not have done without it.

Contents

Figures

Tables

Code Listings

Glossary

Apache Kafka Description mal. 13, 35–38, 40, 69, 70

Clockify Online tool for time tracking. https://clockify.me. 33

comitting Submitting changes to a code repository . https://aloa.co/startup-glossary/terms/code-commit. 34

Discord A social platform for messaging and voice- and video-calls. https://discord.com. 33

Gantt chart A chart displaying tasks or events over time. https://www.gantt.com/. 32, 33

Github Online service for hosting of Git-repositories. https://github.com. 33

Grafana Grafana is a platform for creating and sharing dashboards that combine data from multiple sources, such as logs, metrics, and traces. https://grafana.com. 13, 35, 70, 80

InfluxDB InfluxDB Open Source is a single binary that delivers time series data collection, processing, and analysis. https://www.influxdata.com/products/influxdb/. 13, 35–38, 40, 41, 43, 69, 70

Jira Online software development tool, used for it's digital Kanban board and issue tracking. https://www.atlassian.com/software/jira. 33, 72

Kanban Visual approach to project management. https://www.atlassian.com/agile/kanban. 31

Kanplan A combination of Kanban and a backlog (as used in Scrum). https://www.atlassian.com/agile/kanban/kanplan. 31

Microsoft Teams Online service for collaboration, communication and meetings. https://www.microsoft.com/en-us/microsoft-teams/group-chat-software. 33

MIURA Memory-efficient and Incrementally learning Unsupervised Real-time Anomaly detection for time series data.. iii, v, xvii, 2, 21, 22, 25, 27–29, 35, 40, 43, 44, 69–72, 74–76, 79–81

OneDrive Online platform for storing, sharing and collaborating on files and documents. `https://www.microsoft.com/en-gb/microsoft-365/onedrive/online-cloud-storage`. 33

Overleaf Online collaboration platform for writing and compiling of Latex-documents. `https://overleaf.com`. 33

Python A popular high-level, general-purpose programming language.. 13, 16, 40, 41, 69, 71, 78, 80

PyTorch Open Source framework for machine learning and deep learning `https://pytorch.org/`. 2, 4, 14, 40, 71, 75, 78

RePAD2 Real-Time, Lightweight, and Adaptive Anomaly Detection for Open-Ended Time Series.. iii, v, 1, 2, 4, 21, 22, 35, 43, 69, 70, 72, 74, 79, 80

scikit Popular and open-source machine learning library for Python. `https://scikit-learn.org/stable/`. 40

Signal An encrypted and open-source messaging application for instant messaging. `https://signal.org`. 33

TeamGantt Online tool for creating gantt charts. `https://app.teamgantt.com/`. 33

Telegraf Description `mal`. 13, 35, 37, 38, 69, 70

TensorFlow Open Source framework for machine learning and deep learning `https://www.tensorflow.org`. 4

Webhook Function that allows notification of triggered events `https://www.redhat.com/en/topics/automation/what-is-a-webhook`. 80

Chapter 1

Introduction

This chapter introduces the project, describes our tasks, and outlines the context and objectives it aims to fulfill. It also provides a brief introduction to the authors, the subject area of the project, the limitations imposed, and the subsequent structure of the thesis.

1.1 Project Background

Orange Business [1], formerly Basefarm, is a leading provider of mission-critical IT solutions. Modern IT infrastructures depend extensively on real-time monitoring and analysis of system metrics to ensure optimal performance, security, and reliability. Motivated by this, we undertook the task of implementing real-time and lightweight anomaly detection using the concept of deep learning, based on a state-of-the-art approach called RePAD2 [2], introduced by our supervisor Jia-Chun Lin and her research partner.

Given Orange Business's role in providing mission-critical IT solutions, the implementation of a robust anomaly detection system is vital. Such a system is essential for maintaining the functionality and integrity of critical IT infrastructures, meaning that a high-performance, real-time, and lightweight anomaly detection system would not only benefit Orange Business, but also a wide range of businesses and application domains. The role of an anomaly detection system is to serve as an early warning system, alerting to potential issues, such as unusually high resource usage, sudden temperature increases, or malfunctioning sensors. Therefore, accurate anomaly detection is key to timely address potential issues.

However, the effectiveness of such systems relies on high accuracy, as an approach that creates too many false alarms is not helpful because it constantly disrupts regular operation routines, prompting users to silence such tools [3]. Traditional systems primarily use statistical classification and regression models. While statistical models excel for time series forecasting, they often struggle with adapting

to trends, seasonality and change-points [4]. For this reason, we chose to use deep learning to implement our anomaly detection system.

1.2 Task Description

The objective of this project is to research whether an approach like RePAD2 would benefit from the usage of incremental learning [5] and PyTorch [6]. **Incremental learning** refers to a deep learning model continuously updating its weights based on new incoming data, meaning that it continues to learn from new data without retraining from scratch. **PyTorch** is a deep learning library for Python, and is used to build deep learning models.

Currently, RePAD2 does not adopt incremental learning. Instead, it re-trains its prediction model from scratch whenever the model fails to accurately predict the next data point. Additionally, RePAD2 was implemented using Deeplearning4J [7], which is a deep learning framework based on Java, meant for building and training deep learning models. However, according to Lee et al. [8], PyTorch is considered more time efficient than DeepLearning4J.

In this project, we re-implement RePAD2 using PyTorch, and then extend it by incorporating the concept of incremental learning and name such an implementation as MIURA, which stands for Memory-efficient and Incrementally learning Unsupervised Real-time Anomaly detection approach. Furthermore, we consider three different data scalers for both RePAD2 and MIURA, and then evaluate how these six variants perform on several real-world open-source time series data.

We hope that the results from this project will contribute to the field of anomaly detection. For these reasons, we have chosen to open source all the source code associated with this project. All the code, scripts used in the evaluation and other related material has been made available on GitHub at `https://github.com/patrickjoh/miura`.

1.3 Project Goals

The goals of this project have been divided into three categories: result goals, impact goals and learning goals. Result goals are what the group aims to achieve, in the form of concrete results and deliverables. Impact goals are the reason behind the project and what the group wants the stakeholder to get out of the project. Learning goals are the learning outcomes the group hopes to gain during the duration of the project.

Result Goals

Our solution should achieve accurate real-time detection of anomalies in time series data with minimal false positives, thereby enhancing anomaly awareness

and response capabilities.

- Provide insight into potential improvements to how the individual anomaly detection models work and are trained.
- A model that is able to accurately predict anomalies in a multivariate time series with an F-score and time consumption that is either comparable or better than the original model.
- Provide insight into potential improvements in the RePAD2 approach.
- An approach that is able to accurately predict one point into the future, and accurately detect anomalies

Impact Goals

Our solution should be able to aid companies in accurately detecting anomalies in time series data using minimal resources. Additionally, it should grant greater insight in the use of neural networking in anomaly detection.

- Increased awareness concerning potential anomalies and suspicious activity.
- Better ensure the stability of critical systems and applications through monitoring.
- Enhance the overall reliability and security of the IT infrastructure.

Learning Goals

Our learning goal is to acquire a deeper understanding of deep learning, infrastructure, and agile development methodologies.

- Learn what neural networks are, how they work, and how they are implemented.
- Increased understanding of deep learning and the real world applications.
- Learn and understand how different types of neural networks differ and their applications.
- Gain understanding of how to work with machine learning and deep learning.
- Gain insight and understanding of how deep learning models are developed and evaluated.
- Better understand how agile development methods work and how they can be applied in a real-world context.

1.4 Subject Area

In this thesis we will cover the following subjects:

- Anomaly detection

    - Anomaly detection refers to methods for identifying points in data that deviates from expected behaviour [9].
- Univariate time series
    - A univariate time series refers to observations of one variable taken from a device or a system simultaneously over time.
- Deep learning
    - Deep learning is a subset within Artificial Intelligence (AI) that attempts to simulate the behaviour of the human brain using neural networks to process large amounts of data [10].

1.5 Limitations

Below are the limitations of this project.

- Types of time series - For this project we will focus on univariate time series data to align with the design of RePAD2. Detecting anomalies within multivariate time series is out of the scope of this project.
- Types of anomalies - To reduce complexity, our anomaly detection approach will focus on detecting point anomalies and sequential anomalies. The former refers to single anomalous data points whereas the latter refers to a collection of data point that may appear normal individually but are considered anomalous when observed within a specific time frame. [11].
- Framework - PyTorch will be the primary framework used for implementing our anomaly detection approach. Other frameworks such as TensorFlow will not be considered. This decision is made because PyTorch was shown to provide the best overall efficiency [8].

1.6 Project Group

In this project, Orange Business holds the role of stakeholder with Truls Enstad being the groups main point of contact with Orange. Jia-Chun Lin and Ameen Chilwan are the group supervisors.

The group consists of two students: Patrick Bjørkhaug Johannessen and Magnus Westerheim Johannessen, both from the bachelor programme Digital Infrastructure and Cybersecurity. Although the group members have had similar courses throughout their studies, there are some differences: Patrick has taken additional courses in algorithms and networking, while Magnus has had an extra math course. The team has no prior experience with neural networks or deep learning, which is partly why this assignment was chosen, as it presented a challenge and a great learning opportunity. Both members have had a statistics course, which provides the closest experience the group has in the field.

1.7 Thesis Structure

- ***Chapter 1 - Introduction:*** Description of project background, goals, scope, and thesis structure.
- ***Chapter 2 - Background:*** Essential background for the project.
- ***Chapter 3 - Related Work:*** Existing solutions that are related to our project.
- ***Chapter 4 - Requirements:*** Requirements for our solution, both functional and non-functional.
- ***Chapter 5 - Technical Design:*** Description of technologies implemented into our system and why they were used.
- ***Chapter 6 - Development Process:*** Description of the development process.
- ***Chapter 7 - Implementation:*** Description of how our anomaly detection approach was implemented.
- ***Chapter 8 - Evaluation and Results:*** Performance evaluation of our anomaly detection approach on real-world time series data.
- ***Chapter 9 - Evaluation of Requirements:*** Discussion of how the implementation satisfies the requirements set in Chapter 4.
- ***Chapter 10 - Discussion:*** Discussion about the choices made during the project.
- ***Chapter 11 - Closing Remarks:*** This chapter concludes the thesis by detailing the results and outlining future work.

Chapter 2

Background

This chapter aims to give the foundational knowledge on the theoretical framework that this thesis builds upon.

2.1 Theory

This section briefly introduces the concepts of machine learning, deep learning, neural networks, time series data, and anomalies.

2.1.1 Machine learning and deep learning

Machine learning is a subfield within the overarching study of artificial intelligence (AI). Machine learning was developed as a way for machines to mimic human intelligence by learning existing patterns from data without explicit programming [12] [13]. Machine learning encompasses a broad spectrum of methods covering different applications such as support vector machine (SVM), random forest, k-means, and neural networks [14]. Deep learning as a subfield of machine learning further building upon the goal of machine learning by using deep neural networks. These deep neural networks allow for better flexibility in processing and learning patterns from unstructured data and handling larger dataset, often at the cost of higher complexity and time consumption.

Both machine learning and deep learning cover a wide variety of methods, these various methods also employ different training strategies and can be sorted into supervised learning, unsupervised learning, semi-supervised learning and reinforced learning.

- **Supervised learning:** It involves training models using data that has been labeled with the correct output. This process makes the model recognize patterns between input and output, giving it the ability to correctly classify for new, unlabeled data [15].

- **Unsupervised learning:** It involves training models using unlabeled data, making the model recognize patterns and classify data without the need for human intervention [16].
- **Semi-supervised learning:** a combination of both supervised and unsupervised learning. In this approach, the model is trained using a mix of labeled and unlabeled data [17].
- **Reinforced learning:** It involves training a model to reach an end goal by autonomously attempting various actions. Through a process of trial and error, it receives feedback from the environment, which helps the model train itself to correctly complete tasks [18].

2.1.2 Neural networks

Artificial neural networks are made to mimic the way humans learn by emulating the biological neural network found in brains, through the use of mathematical functions. An artificial neural network consists of neurons and synapses, as illustrated by Figure 2.1. As data traverses the neurons and synapses, it is put through several mathematical functions so that it can adjust and fit the data according to what the model is trained for [19] [20].

There are several different types of neural networks, with *feed forward neural networks* being one of the most fundamental types. In this network, neurons are separated into layers connected by synapses. There are three different types of layers: the input layer, hidden layer and output layer. Within one network there is one input layer and one output layer, but there can be several hidden layers.

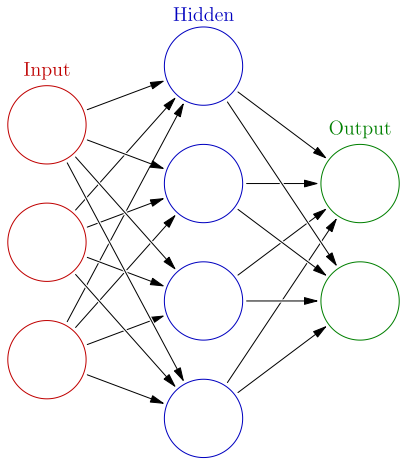


Figure 2.1: Feed forward neural network [21]

When data enters the input layer, it traverses the synapses connecting it to the hidden layer. While traversing a synapse the data is multiplied by the synapses set weight, which is a number that can either be positive or negative. The Weight can

be abstracted as a part of the neuron as it is specific to the connected neuron [22]. When data reaches the artificial neuron within the hidden layer, it will be applied three steps: summation of incoming data, adding of bias (constant specific to a neuron), and inputting the result in an activation function [20].

During training, the neural network continually adjusts its weights and biases to better fit the expected output through a process called backpropagation. Backpropagation works by comparing the results of the network with the expected results and adjusting accordingly [23].

Recurrent neural networks (RNN)

A problem with feed forward networks is that they are not designed to consider past inputs, making them non-ideal considering sequential data such as time series [24]. Recurrent networks tackle this weakness by including "memory" of previous hidden states in the calculation of the final output as the hidden state carries information about last input, hidden state being the processed input.

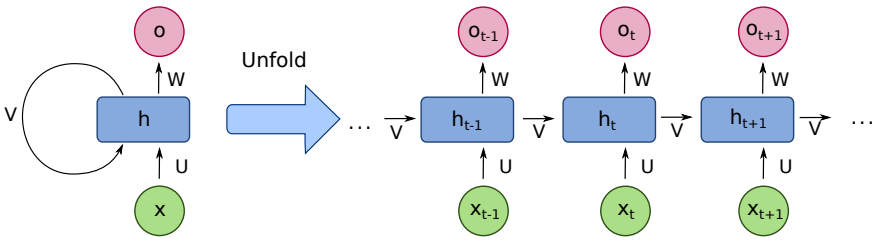


Figure 2.2: Recurrent neural network [25]

The architecture of a recurrent neural network is similar to the feed forward neural network in that it has three different types of layers, the main difference is how it handles the hidden layer. The hidden layer in a recurrent neural network integrates a sort of "memory" that allows it to keep track of what has previous been inputted. It does this by making each node take two different inputs: new input and the latest hidden state. The recurrent neural network does this by looping the last hidden state back into the node as shown in Figure 2.2. the loop repeats for k times signifying how many previous time points taken into consideration of calculating $k+1$ [26][27].

To train the recurrent neural network uses a version of backpropagation called backpropagation through time (BPTT). This works by calculating the error of each timestep k, progressively working through each layer and multiplying the results [28]. This introduces the problems of vanishing and exploding gradients. As the gradient exponentially gets larger or smaller updating the weights with this gradi-

ent will result in a weight that is non-representative and the learning function fails [29][30].

Long Short-Term Memory (LSTM)

Besides the problem of the vanishing and exploding gradients, recurrent neural networks are not good at retaining long term memory. To solve these problems Hochreiter et al. [31] propose the Long Short Term Memory (LSTM) model.

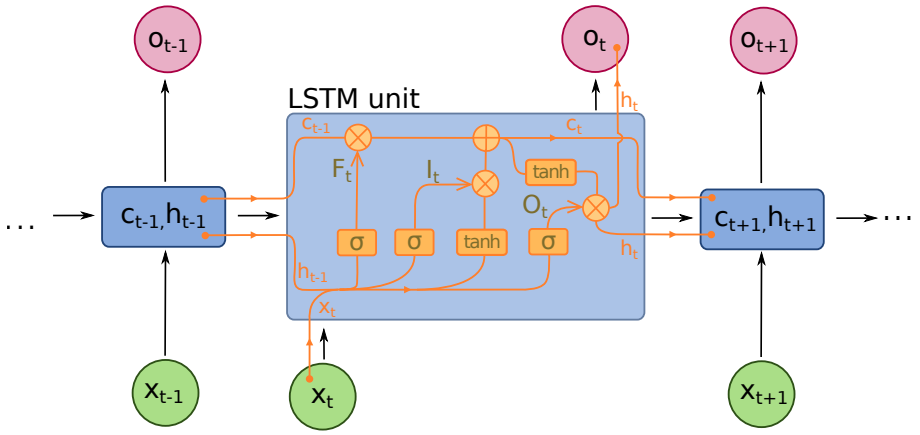


Figure 2.3: Long Short Term Memory model (LSTM) [32]

The LSTM model has the same overarching architecture as the recurrent neural network, in that it has multiple layers of the same node where information is looped and passed on to another iteration of the node. However it increases the complexity by adding another state where important information is stored long term, called the cell state, marked as C in Figure 2.3. The LSTM model also introduces several *gates*. These gates serve the purpose of filtering out old irrelevant information from the cell state, while also determining what information from the new input (X) and hidden state (h) should be stored [33].

Three gates are employed in the LSTM model: the forget gate, input gate and output gate. These three work together with the cell state (C), hidden state (h), and the current input (X) to give accurate output while avoiding the problems of vanishing and exploding gradient.

When new data and the previous hidden state enter the LSTM instance, it is first used by the forget gate to decide what information should be deleted from the cell state. It does this by combining the new input and the previous hidden state, then running them through a sigmoid activation function that will output a number between 0 and 1. The value of 0 means full removal, and 1 means no removal [34] [31].

The input gate then decides what new information will be stored in the cell state,

which involves two stages. First, the combined hidden state and the input is put through a sigmoid function that will decide what data in the cell state that should be updated. Then input and hidden state data is run through a tanh activation function to create a vector of values that could be added to the cell state. The output of the sigmoid and tanh functions are then multiplied to create the data that is added to the cell state [33].

The output gate is the part of the LSTM model that determines which parts of the cell state will be output. This is done by running the combined new input and previous hidden state through a sigmoid funtion, then multiplying the results to information from the cell state that has been ran through a tanh function. The result of this multiplication will be the output and the new hidden state [33].

2.1.3 Time series data

Time series data is a type of data where a variable is mapped over a fixed axis of time, meaning time is measured in constant increments, as seen in Figure 2.4 [35]. The variable recorded over time is refereed to as a feature. Any time series associated with only one feature is called univariate time series, while any time series that is associated with more than one feature is called multivariate time series [36]. Time series data is used within a variety of fields such as finance, healthcare and system monitoring, and it is useful since it provides deeper insight into fluctuations in data and its correlation with time.

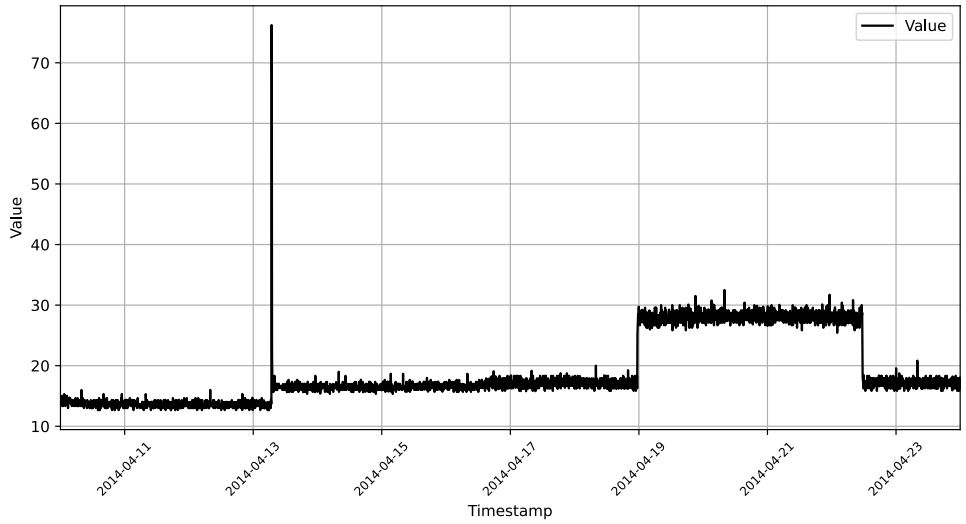


Figure 2.4: Example time series

Time series data is special as it can contain several attributes, properties and patterns that are necessary to understand when analyzing such data. Patterns in time series are usually separated into three categories: trend, seasonal and cyclic. Trend

refers to long term increase or decrease in value over time. Seasonality is when the value of a data point is affected by the specific moments in time, such as day of the week or certain periods within a year. Cycles are when data goes through recurring periods of increase or decrease, but these periods of fluctuation are not fixed like seasonal patterns [37].

Another property of time series that one must take into consideration is whether or not a time series is stationary. The data in a stationary time series exhibits constant statistical attributes such as mean and variance, meaning the data has no trend or seasonal components [38] [39].

2.1.4 Anomalies and anomaly detection

Anomalies are data points that deviate significantly from what is considered normal and expected behaviour [40]. Anomalies happen for a variety of reasons depending on the nature of the original data. An example could be malicious behaviour towards a network, causing fluctuations in traffic. Anomaly detection refers to methods of identifying anomalies, it is a field of study that has been of great interest as it allows for better monitoring and possibly early detection of problems or attacks. In fields where early detection is critical for managing the consequences of incidents, automatic anomaly detection plays a crucial role.

For a system to determine what is considered an anomaly, it needs a set definition of what consolidates normal data, which will vary depending on the chosen method of anomaly detection and the data being analyzed. However, it is generally done by defining or calculating a set baseline that determines what could be classified as normal data. Within time series anomaly detection, this is generally done by analyzing historical data and forecasting future data points. As stated by Palmieri et al. [41], defining what is considered normal behaviour can only be decided by historical data.

Using historical behavior to predict future data points and calculating an anomaly score, along with a set threshold, is a common method for identifying anomalies. The anomaly score indicates the likelihood of a data point being anomalous, while the threshold determines which scores fall within acceptable values.

Time series anomaly detection differs from regular anomaly detection as data is indexed in a sequential order. This adds another dimension to anomaly detection; it is no longer strictly spatial but also temporal, meaning that anomalies must be identified based not only on data values but also on their order and changes over time.

Anomalies in time series can be categorized into four different types:

Point anomalies: individual data points that significantly deviate from the rest [11].

Collective anomalies: data points that would not be considered anomalous if they were to appear individually [11].

Sequential anomalies: an anomaly spanning more than one individual data point [42].

Contextual anomalies: data points that are considered anomalous only within a specific context. An example of this would be that the amount of visitors at IKEA spike towards the end of summer, if this spike were to happen at any other time of the year it would be considered anomalous, but since it is the end of summer it is normal behaviour [11].

2.2 Technology

This section will briefly introduce technologies and applications used in the development of this project.

Apache Kafka [43] is a popular event streamer meaning it is used for receiving, processing and publishing data, as a "broker" since it is between the source and the destination of the data. Apache Kafka is used for high-performance data streams, streaming analytics and data integration[43]. Kafka enables multiple producers such as IoT devices to send data to a Kafka broker. The broker then stores the event data in "topics" to sort different types of data. The data is then sent to the destination, a "consumer" [44].

Telegraf [45] is a modern day data handler used for collecting and sending various events and metrics from databases, systems and IoT sensors [45]. A practical use-case for Telegraf is for example to collect data from an IoT sensor then forwarding it to a database such as InfluxDB [46]. Telegraf is written in Golang and requires no additional dependencies. It is also extremely lightweight allowing it to run on most devices. Telegraf is supported by a vast variety of different technologies and datatypes allowing for easy implementation. If the required datatype is not supported it is possible to further develop plugins specifically to suit use-case.

InfluxDB [46] is an open source time series database developed by Influx data. A time series database is a database meant for storing time series data with high performance, making it able to ingest large amounts of data in short spans of time, which is useful when storing and analyzing metrics data or other monitoring data [47]. InfluxDB is managed by users through the query language Flux allowing for easy data manipulation [47].

Grafana [48] is an open source visualization tool that is often used for monitoring, logging and other types of visualization. Besides visualizing data, Grafana also allows for built in alerting should it discover a problem [49].

PyTorch [6] developed by Meta AI [50], is an open source framework for Py-

thon built for the development of deep learning models, based on the older torch library [51]. PyTorch is widely used as it helps streamline the development process of deep learning models, and integration of both CPU and GPU processing, while keeping high efficiency [52].

Chapter 3

Related Work

This chapter will introduce existing anomaly detection approaches for time series data, then further discuss their strengths and weaknesses as compared with our approach.

Anomaly detection is a field that has been of great interest these past couple of years within a wide variety of fields: intrusion detection, health monitoring, industrial machines, etc. Current time series anomaly detection approaches can be separated into three categories: statistical, machine learning, and deep learning anomaly detection.

3.1 Statistical methods

Statistics have a long-standing history in the field of anomaly detection, with statistical methods being among the oldest techniques used [11][53]. Statistical methods vary greatly in how they handle data distribution and determining what is considered anomalous behaviour. A generalized description could be: techniques that model the underlying statistical attributes of data, then test if new data adheres to the model [40].

3.1.1 Arima Based Network Anomaly Detection

Yaacob et al. [54] suggests the use of the statistical forecasting technique: Auto-Regressive Integrated Moving Avarage (ARIMA) for detecting attacks on network traffic through the use of anomaly detection. The approach uses the ARIMA method to predict what is considered normal data flow and uses this prediction compared to the actual data point to calculate distance. If the distance calculated between the actual data point and the predicted is above the threshold, an anomaly is detected.

The proposed system implements a static threshold, meaning domain experts need

to have prior knowledge of system behaviour to determine acceptable values for data. Acceptable values vary depending on the data being processed meaning the threshold has to be manually adjusted to fit the system. As data changes, what is considered an acceptable value might change, meaning the static threshold has to be manually updated. A static threshold also results in limiting the effectiveness of the approach on shorter more erratic time series. ARIMA is also a parametric method which means it would require proper parameter tuning.

3.1.2 A Real-Time Temperature Anomaly detection Method for IoT Data

Liu et al. [55] suggest the use of statistical methods for temperature anomaly detection on IoT devices in medical refrigerators. One of the key points of the approach is the use of the smoothed Z-score algorithm to predict the next data point, and to augment the threshold that determines what is considered an anomaly to dynamically fit historically acceptable temperature variations.

The approach consists of two phases: testing and initialization of data and anomaly detection. The testing and initialization phase focuses on testing if the underlying data is stationary and creating the time lag used for prediction and threshold setting. The approach makes the assumption that the underlying data is stationary and bases both prediction and threshold on that assumption, meaning that it would not work well on datasets that display non-stationary attributes.

3.1.3 Luminol

Luminol [56] is a publicly available anomaly detection library built for Python. The library includes the possibility of choosing several different anomaly detection methods making it able to be fit towards different use-cases. The primary function of Luminol is to take a given time series and then calculate an anomaly score based on the likelihood of a certain time window being anomalous. However, for the library to be effective, the user has to determine a predefined threshold or the user has to review if the data points are deemed as anomalous.

3.1.4 Automatic Anomaly Detection in the Cloud Via Statistical Learning

Many different anomaly detection methods have been researched and proposed, but none have been suitable for social networks. The reason for this is that the underlying data of social networks have a high amount of trend and seasonality, which many anomaly detection approaches struggle to handle. To solve this problem Hochenbaum et al. [57] introduced two novel methods: S-ESD and S-H-ESD that tackle the seasonal and trend components in time series data.

The two proposed methods use decomposition to separate the seasonal, trend and residual components of a time series. After the decomposition, S-ESD uses

the extreme studetized deviate (ESD) test to detect anomalies from the decomposed data stream. On the other hand, the S-H-ESD method uses Mean Absolute Deviation (MAD) to detect anomalies as it is more robust against larger amounts of outliers at the cost of higher computational complexity. The approaches show promising results in detecting outliers in the presence of seasonal and trend components. However, the methods are parametric meaning they would require tuning, which requires iterative testing and potentially domain knowledge to achieve best results.

3.1.5 Anomaly Detection in Streams with Extreme Value Theory

Siffer et al. [58] proposes a solution using extreme value theory to automatically set thresholds in anomaly detection, as previous techniques have relied on manual adjustments or assumptions about the underlying data.

The approaches uses extreme value theory implemented with the peak over threshold methods. Two algorithms are introduced, the SPOT algorithm for streaming data with stationary attributes and the DSPOT algorithm for data that could potentially be subject to concept drift. The approaches starts initialization with a high threshold, when an anomaly exceeds this threshold the approaches uses the the Generalized Pareto distribution on the excess above the active threshold. Using this allow for the approaches to adapt the threshold to stay within a certain percentage of anomaly. But since the approaches rely on anomalies to calibrate, as anomalies often are scarce it could potentially take a long period of time for the approaches to calibrate properly, depending on the data stream.

3.2 Machine learning approaches

Machine learning is a term that covers a wide variety of different techniques and approaches, all with the same goal of automating the process of acquiring knowledge through the use of examples. In anomaly detection this becomes the practice of using machine learning models that have been trained to recognize the difference between abnormal and normal points in data [59].

3.2.1 A novel unsupervised anomaly detection for gas turbines

Anomaly detection for gas turbines is of great interest to ensure safe operating and preventing costly maintenance, but most previous approaches have used classification-prediction models trained on large samples of labeled data. The lack of labeled anomalous real-life datasets makes an unsupervised anomaly detection approach of high interest. Zhong et al. [60] proposes an approach that utilizes the isolation forest algorithm [61] to detect anomalies.

The proposed method consists of three steps: separating data into groups, preliminary anomaly detection, and precise anomaly detection. The preliminary anomaly

detection is more general, and used to filter out the groups that are deemed as normal. Data that is deemed as anomalous go through a second isolation forest method that more precisely detects anomalies. The approach function in an unsupervised manner without assuming the underlying distribution of the data. However, as the system works in batches of historical data, it will not be able to detect anomalies as they happen in real time.

3.2.2 Generic and Scalable Framework for Automated Time-Series Anomaly detection

The EGADS framework proposed by Laptev et al. [62] was created because current state of the art anomaly approaches suffer from a lack of scalability, use-case restriction and difficulty of use.

The EGADS framework is highly modular and consists of three core modules: time series modeling, anomaly detection, and alerting module. The specific method used within a module can vary based on the needs of the end user. This is because the performance of each method is highly case specific. Because of how the framework is designed, it requires the time series module to model the data stream so that it can feed the predicted values to the anomaly detection module, thereby adding computational overhead.

3.2.3 Network anomaly detection through nonlinear analysis

Networks today are more susceptible to attacks than ever, making analysis of network traffic increasingly important. Palmeri et al. [41] proposes an anomaly detection approach that builds upon nonlinear analysis through machine learning and statistical methods to detect recurring patterns in data traffic, allowing for automatic detection of specific attack patterns.

The approach uses recurrence quantification analysis (RQA) to detect the recurring non-stationary attributes of the incoming data stream, such as %REC (recurrence rate), %DET (determinism, indicating predictable behaviour), entropy (measures complexity) and LAM (signifying constant behaviour). The system uses a Support Vector Machine (SVM) trained to classify data points as normal or anomalous. This classification is based on the non-stationary attributes identified through RQA. As the SVM method is trained offline, it will not be able to detect new attack patterns. The RQM also requires parameter tuning, which requires expert knowledge.

3.2.4 Event Detection in Marine Time Series Data

Within the marine domain, detection of special events that deviate from the norm is often more interesting than the norm itself. Oehmecke et al. [63] suggest using the Local Outlier Factor [64], an unsupervised anomaly detection algorithm, to identify special events.

To improve the effect of the local outlier factor (LOF) detection, data is pre-processed with noise reduction and dimensionality reduction. Anomalies are identified by calculating the distance between a data point and the surrounding neighbours, where only the points with the highest score are identified as outliers. But this approach requires the specific tuning of a value k to determine how many neighbours should be taken into consideration when calculating anomaly score, as the optimal value could be dependant on the dataset. Another limitation of the approach is the requirement for data pre-processing and number of calculations, making the approach less suited for a real-time approach as this will cause latency and delay.

3.3 Deep learning approaches

In recent years, deep learning has made its way into the field of anomaly detection because of its ability to learn and categorize complex patterns in data through the use of deep neural networks. Deep learning is able to provide better performance without assuming the underlying statistical distribution. Traditional machine learning approaches have struggled with understanding data that has higher complexity, something deep learning excels at [65]. Today there exists a variety of different deep anomaly detection methods that are suited and adapted to best fit the environment it is designed for.

3.3.1 USAD: UnSupervised Anomaly Detection on Multivariate Time Series

Orange Business has previously used anomaly detection that trains in a supervised manner, meaning the approach requires domain experts to label datasets and then feed the labeled datasets to the model for it to learn. As IT-operations at Orange has grown larger and more complex, such methods are no longer feasible leading to the creation of USAD [66].

The USAD approach employs an encoder-decoder architecture together with adversarial training to perform unsupervised multivariate anomaly detection. Training of the USAD approach is done in batches offline, and consists of two phases: autoencoder training and adversarial training. The approach could be scheduled to regularly retrain, but it will not result in the same adaption to unforeseen changes in data as a fully automatic online approach using a feedback mechanism.

3.3.2 DeepAnt: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series

Traditional anomaly detection methods, based on distance and density, often fail to identify periodic and seasonal point anomalies. This limitation leaves much to be desired in anomaly detection for IoT devices. To tackle this problem, Munit et al. proposes the deep learning-based solution named DeepAnT [67].

Traditional anomaly detection methods, based on distance and density, often fail to identify periodic and seasonal point anomalies. This limitation leaves much to be desired in anomaly detection for IoT devices.

The approach is comprised of two modules: time series predictor and anomaly detector. The time series predictor uses a convolutional neural network (CNN) to predict one data point into the future. The anomaly detector takes the predicted data point and determines an anomaly score through the use of euclidean distance between the predicted and actual data point. The approach is able to handle seasonal anomalies as long as they are represented in the training set provided the model. The model is trained in an offline manner, meaning it would require regular offline retraining to adapt to new data. In the paper, it is also stated that the approach will only handle up to 5% anomalous data in the training set.

3.3.3 Incremental learning of LSTM-autoencoder anomaly detection in three-axis CNC machines

As CNC machines have developed to become more autonomous, there is an increased demand for automatic anomaly detection. Existing machine learning and deep learning approaches for CNC machines are flawed because training from scratch requires a large amount of data. Although transfer learning has been tested to address this issue, it has resulted in lower accuracy. To solve this problem, Li et al. [68] propose combining transfer learning with incremental learning specifically for CNC machines. This approach aims to minimize training time and adapt more effectively to current anomalies.

The approach uses an architecture consisting of several LSTM-autoencoders. The system starts with a main strong learner trained on old data, as new data arrives the approach adds multiple weak learners that generalize the new incoming data making the model able to catch nuances in new data. The complexity of the approach will continue to increase, as it takes on more weak learners, adding to its computational intensity. Since the model is supervised, it will never be able to provide real time feedback. When new data arrives, it will have to be labeled before the anomaly detection model is able to use it for training. This also creates the problem of always having to label data as it arrives.

3.3.4 Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning

In the research of deep learning for anomaly detection in time series, previously proposed models have mostly consisted of recurrent neural network based models. Wen et al. [69] propose a convolutional neural network (CNN) based anomaly detection approach that utilizes transfer learning.

The approach is able to think of a time series data stream as a one dimensional image and analyze the image for outliers. Regular snapshots are taken of the current

time series and fed to the model for it to detect potential anomalies. The approach described uses a model that is pre-trained on a large labeled synthetic data giving it good generalization. Then the generic model weights are transferred and further trained on a smaller case-specific labeled dataset. This allows for good case specific knowledge of anomalies while keeping real labeled data to a minimum. Since the model is trained offline, without a feedback mechanism initiating re-training, it would not be able to adapt to sudden changes in the data without being taken offline for re-training on new labeled data.

3.3.5 RePAD and RePAD2

Lee et al [70] proposed the RePAD approach for anomaly detection in time series data, unsupervised and fully online. RePAD employs an LSTM model that uses a short historic time window of three data points to predict one data point in the future. RePAD calculates the Average Absolute Relative Error (AARE) value [71] from the predicted and actual values. RePAD starts calculating the threshold as soon as three AARE values have been calculated. Since RePAD uses all previous AARE values to calculate the threshold, it will become progressively more computationally intensive and run into a resource exhaustion problem.

RePAD2 proposed by Lee et al. [2], a predecessor to the original RePAD approach, is built to handle the resource exhaustion problem that the original algorithm suffered from. To solve this problem, the RePAD2 approach utilizes a sliding window that limits the amount of previous AARE values taken into consideration when calculating the threshold, effectively solving the resource exhaustion problem. However, since RePAD2 retrains the LSTM model when a potential anomaly is detected, it looses the previously held information on what the previous data points looked like, meaning it will not be able to see the data in a larger context.

3.4 Summary

The approach MIURA that is introduced in this thesis, is based on the RePAD2 approach [2], and is fully unsupervised and online. Both RePAD and MIURA make no assumption of the underlying distribution of data, unlike many statistical methods. This makes the approaches more flexible and capable in terms of adapting to a data stream, without needing the stream to have certain attributes. Both MIURA and RePAD2 have a dynamic self adjusting threshold, meaning that the model does not require any prior knowledge of the time series. Since the approaches are unsupervised, they will not need to train on data labeled with the correct output. As stated by Wen et al. [69] and Audibert et al. [66], real life labeled data is often scarce, making an unsupervised approach highly attractive as they do not require training on labeled data beforehand.

The approaches are also able to learn in a fully online manner, meaning they do not require being taken offline to adapt to new data patterns. The models have

an implemented feedback mechanism that enables automatic calibration to new data patterns.

The MIURA approach introduces incremental learning to see if it can improve detection results, and better detect anomalies in seasonal data, as this is something RePAD2 struggles with. As seen in the solution proposed by Li et al. [68], the incremental learning approach causes the system to become progressively more computationally intensive. This could be solved by keeping the weights from last time the model trained, and training that model on the new data. This approach leads to no increase in resource usage or complexity.

Chapter 4

Requirements

This chapter will define the requirements for our solution, which are divided into two types: functional and non-functional. Functional requirements refer to the functionality and how the solution interacts with internal and external components. In simpler terms, they describe what the solution is supposed to do. On the other hand, non-functional requirements refer to the properties and qualities of the solution.

4.1 Functional Requirements

F1: Store data persistently
A database must be implemented to persistently store incoming data. This database will maintain both incoming time-series data and detected anomalies as reported by MIURA. Storing data persistently is important in case something should happen to MIURA while running, or in the cases where MIURA for some reason will be taken offline.

F2: Receive time series data in real-time
There must be a way to receive time-series data from either a dataset or a sensor. The system must be able to receive and move this data in real-time, which means that it must be able to run continually, receiving and making data available to MIURA in real-time.

F3: Access and process received data
There needs to be a way for MIURA to continually fetch and access new data. The approach is meant to stay online, process incoming data, wait for new data to arrive, process new data.

F4: MIURA must be able to incrementally train an LSTM model
MIURA must be able to keep the current LSTM model and update it with further training. This is what makes the training incremental. This differs from RePAD2 [2], which have to train a new LSTM model every time.

F5: MIURA must be able to report anomalies when detected
There needs to be a way for MIURA to report detected anomalies, and for the reported anomalies to be stored persistently afterwards. This data can be used for tracking the anomaly and for visualizing the detection afterwards. This needs to be sent and stored somewhere persistently in order to be usable.

F6: Visual monitoring of detected anomalies in real-time
There should be a way to visually monitor the incoming data, and reported anomalies. This is helpful for better understanding and recognizing the patterns in the data, as well as visually determining where and why an anomaly was detected. This clarity can also help identify whether an anomaly is a false positive.

4.2 Non-Functional Requirements

NF1: MIURA must be able to process data faster than it arrives
In order for the anomaly detection to not halt behind real-time, MIURA needs to receive data, process that data and report anomalies faster than new data arrives.

NF2: MIURA should be efficient enough to run on a standard consumer laptop
The anomaly detection should not require any dedicated or specialized hardware to run, and it should not become increasingly demanding of resources. It should generally be able to fulfill functional and performance requirements while running on consumer hardware. This enables multiple instances of MIURA to be deployed in a production environment while still keeping hardware cost, electricity needs and heat generation low.

NF3: MIURA must be able to detect anomalies in various datasets
The proposed anomaly detection system should not be domain specific, meaning it is able to generalize and adapt itself to different contexts and types of time series data, from a variety of sources.

Chapter 5

Technical Design

This chapter will explain the design and functionality of our solution. The chapter will cover the different components of MIURA as well as the services it depends on, and how the anomaly detection results can be monitored and visualized in real time.

5.1 System Architecture

Figure 5.1 illustrates the system architecture that MIURA is a part of. It consists of four different components:

- Event Streamer
- Intermediary Data Handler
- Database
- MIURA

New times series data is first received by the event streamer. Data is then fetched and sent to the database with the intermediary data handler. Any data point reaching the database will be immediately processed by MIURA in real-time. When MIURA detects any anomalous data point, it immediately saves the anomalous information into the database, making that data available to the monitoring dashboard. In the following subsections, each component in this system will be introduced and explained in further detail.

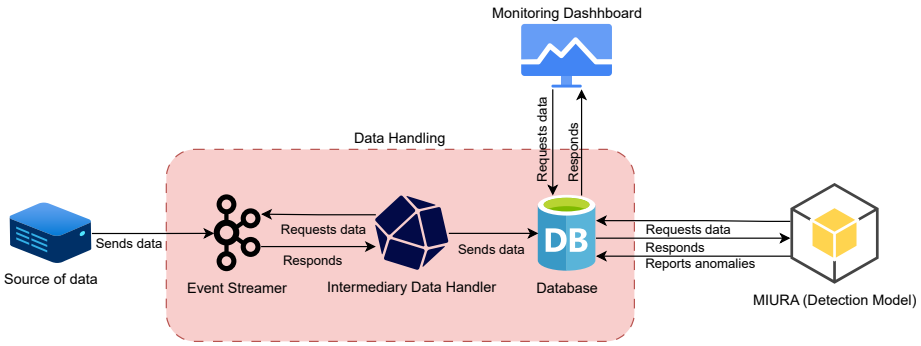


Figure 5.1: A system architecture for time series data handling and anomaly detection

5.1.1 Event Streamer

This component is responsible for receiving time series data from an external source and making this data available to the database. The event streamer acts as a buffer for the database and is capable of handling huge amounts of incoming and outgoing data, in real-time. It does not collect data itself, but passively receives data from the data source, which can be any sensor or system. The event streamer is referred to as a broker as it accepts data from the data source and give data to a consumer, which is the intermediary Data Handler.

5.1.2 Intermediary Data Handler

The responsibility of this component is to consume data from the event streamer and feed it to the database. Data is periodically requested from the event streamer, collected in batches, and sent to the database. The size of these batches are a configurable part of this component, and the optimal size for one such batch depends on how many values each data point contains. Data could have been sent directly to the database, but the number of write and read requests may have rendered the database unresponsive.

The intermediary data handler works in combination with the event streamer to lessen the overall load on the database by reducing the number of incoming write requests. For each write request, the database needs to unpack and authenticate the request and then write the contents, increasing total overhead. Batching multiple data points in a single write request allows the database to efficiently process contents while only having to unpack and authenticate a single request.

5.1.3 Database

The purpose of the database is to persistently store all incoming time series data points. A time series data point always contains a timestamp with time and date

for when it was created, and one or more accompanied values that are conventionally either integers, floats or strings [72]. The database receives this data from the intermediary data handler, and then makes it immediately available to MIURA. Anomalies are then reported back to the database by MIURA and made available to the monitoring dashboard. This data is kept as time series data, each data point keeping original values, timestamp and sorted in correct chronological order, ensuring that all data reaching MIURA is in the original order. It is very important that time series data is kept in original order as changing the order also changes the patterns in the data, making accurate analysis on that data impossible.

5.1.4 MIURA

Code listing 5.1 illustrates the pseudo code of MIURA. When MIURA starts, the current time point T is set to 0. To predict the next data point, MIURA uses the three most recent data points, and saves D_{T-2}, D_{T-1} and D_T at time points 0, 1 and 2. When T is 2, MIURA is able to initialize and train the LSTM model, denoted by M, with the three aforementioned data points. The LSTM model is then used to predict the next data point $\widehat{D_3}$ (See lines 7 and 8 in Figure 5.1). When T advances to 3 and 4, MIURA updates the LSTM model and predicts $\widehat{D_4}$ and $\widehat{D_5}$, respectively (See lines 10 and 11). When T equals 5, MIURA calculates $AARE_5$ based on Equation (5.1).

$$AARE_T = \frac{1}{3} \sum_{y=T-2}^{T} \frac{\left| D_y - \widehat{D}_y \right|}{D_y}, \quad T \geq 5 \tag{5.1}$$

D_y is the observed value at time point T, and $\widehat{D_y}$ is the predicted value at time point T. MIURA always calculates $AARE_T$ using the three most recent absolute errors, of which the three most recent pairs of D_T and $\widehat{D_T}$ are each used. The resulting $AARE_T$ is the mean of these values. When T advances to 6, MIURA calculates $AARE_6$, and updates the LSTM model to predict $\widehat{D_7}$ (See lines 14 to 16). When T further advances to 7, MIURA calculates $AARE_7$, and then calculates the adaptive detection threshold Thd using Equation (5.2), based on the Three-Sigma Rule [2] [57] (See line 21). To prevent resource exhaustion, MIURA always uses the 8064 most recent AARE values to calculate Thd. If the total number of historical AARE values is less than 8064, all AARE values will be used to calculate Thd.

$$Thd = \mu_{AARE} + 3 \cdot \sigma, \quad T \geq 7 \tag{5.2}$$

In Equation (5.2), μ_{AARE} is the average AARE, and σ is the standard deviation from the average. After calculating the detection threshold Thd, MIURA uses it to decide whether or not the current data point is anomalous. If $AARE_T$ is lower than the threshold (See line 23), D_T is not considered anomalous and the LSTM model will not be updated. However, if $AARE_T$ exceeds the threshold (See line

24), it might indicate either a change in the data pattern, or that an anomaly have occurred.

When $AARRE_T$ exceeds the threshold, MIURA creates a copy of the LSTM model M, denoted by MC, and updates MC (See line 25 and 26), trying to adapt to a potential change in the pattern. MC is then used to re-predict D_T and a new $AARE_T$ value is calculated. If this new $AARE_T$ value is lower than the threshold (See line 31), MIURA does not consider D_T anomalous. On the other hand, if the new $AARE_T$ value is still higher than or equal to the threshold, meaning that the updated LSTM model is unable to predict D_T, then D_T is immediately reported as an anomaly. When this happens, MC is not saved and *flag* is set to false, signalling that the LSTM model M must be updated at the next time point (See line 40). Note that instead of directly replacing the LSTM model, MIURA creates a copy of its current LSTM model, referred to as MC in Figure 5.1. This copy is then updated and used to predict. If the resulting $AARE_T$ value is lower than the threshold, the original LSTM model M is replaced by MC.

```
# Let T be the current time point, starting from 0; Let flag be True;
While time has advanced {
    Collect data point D_T;

    if T >= 2 and T < 5 {
        if T = 2:
            Intialize LSTM model by taking D_{T-2}, D_{T-1} and D_T as training data;
            Let M be the resulting model and use M to predict D_{T+1};
        else:
            Update M by taking D_{T-2}, D_{T-1} and D_T as training data;
            Use M to predict D_{T+1};

    } else if T >= 5 and T < 7 {
        Calculate AARE_T based on equation 5.1;
        Update M by taking D_{T-2}, D_{T-1} D_T as training data;
        Use M to predict D_{T+1};

    } else if T >= 7 and flag == True {
        if T != 7 {Use M to Predict D_T;}
        Calculate AARE_T based on Equation 5.1;
        Calculate Thd based on Equation 5.2;

        if AARE_T <= Thd {D_T is not considered an anomaly;}
        else {
            Copy M and let the model be MC
            Update MC by taking D_{T-3}, D_{T-2} and D_{T-1} as training data;
            Use MC to repredict D_T;
            Re-calculate AARE_T based on Equation 5.1;
            Re-calculate Thd based on Equation 5.2;

            if AARE_T <= Thd {
                D_T is not considered an anomaly;
                Let M be the updated MC;
                Let flag be True;
            } else {
                D_T is reported as an anomaly immediately;
                Let flag be False;
            }
        }
    } else if T >= 7 and flag == False {
        Copy M and let the model be MC;
        Update MC by taking D_{T-3}, D_{T-2} and D_{T-1} as training data;
        Use MC to repredict D_T;
        Calculate AARE_T based on Equation 5.1;
        Calculate Thd based on Equation 5.2;

        if AARE_T <= Thd {
            D_T is not considered an anomaly;
            Let M be the updated MC;
            Let flag be True;
        } else {
            D_T is reported as an anomaly immediately;
            Let flag be False;
        }
    }
}
```

Code listing 5.1: Pseudo code of MIURA

5.1.5 Monitoring Dashboard

Processing data and detecting anomalies in real-time is the main goal of this bachelor project, but being able to verify and monitor data in real-time is a great advantage when trying to determine the cause of each reported anomaly. The purpose of this monitoring dashboard component is to clearly visualize the target time series data, while also making it clear where anomalies were detected. The raw data is queried from the database and then visualized in a periodically updated Cartesian plane, where x-axis represents time and y-axis represents the value of that data point in time. Additionally, anomalies are clearly and automatically annotated with a vertical line at each point in time where an anomaly was detected.

Chapter 6

Development Process

This chapter will detail how the group structured itself and worked together. The chapter will complement the project plan in Appendix A, and build upon how the process we planned to follow was actually followed throughout the project. How the project work was documented throughout the project will also be detailed.

6.1 Development model

As the group had no previous experience within the fields machine learning or anomaly detection, the necessary steps involved in the development were highly uncertain. Because of this, the group decided to follow the agile development method Kanplan[73]: a combination approach consisting of Kanban's visual approach and the backlog from SCRUM. This approach is a near pure Kanban approach while avoiding the overwhelming "To-Do" list ensuring that prioritized tasks remain at the forefront.

Following agile methods allowed the team to continually adapt the development process by adding tasks and re-prioritizing on the fly. This approach was necessary as the team progressively gained a deeper understanding of the subject area and the steps needed to complete the project. Kanplan also made it easier to break down larger tasks into smaller, more manageable ones.

6.2 Collaboration

The team only consists of two members, so it was decided that, for most workdays, we would be meeting on campus and work together for a standard 8-hour day. This approach made it sufficient to have one weekly meeting to summarize progress. Although daily meetings at the start of each day to discuss the plan were considered, it were ultimately deemed unnecessary and an additional overhead.

This was because both team members worked in close proximity and frequently discussed their tasks throughout the day.

Division of work

After the initial learning period at the start of the project, one of the supervisors suggested splitting the project into two parts: implementation and literature review. This division provided the team with two distinct tasks, making it easier to delegate work.

6.3 Documentation

Documentation is an important part of project work as it ensures that everyone is aware of the progress being made while, maintains consensus on what has been discussed, and saves time by preventing the need for backtracking. **Meeting minutes**

Before meetings with our supervisors, the group wrote weekly status reports containing progress, challenges, and what was to be discussed in the following meeting. The writing of these status reports was done at the end of the week and served as an internal meeting where the group discussed what has been done and what is to be done the following week. During every meeting with supervisors and stakeholders, meeting minutes were written. After the meetings, we discussed internally what was said during the meeting and how we will move forward based on the received feedback. This was done while correcting, and if needed adding to the notes.

Time management

Each group member was required to document hours spent on the project. Hours spent on the project were to be labeled and provided with a short description of what the time was spent on. While the group wrote the project plan, in Appendix A, a Gantt chart was made to illustrate what amount of time should be delegated to each activity. This worked as to monitor continuous progress and time spent on the project, and made the group more aware if certain tasks were taking too long and the potential of falling behind schedule.

6.4 Routines

6.4.1 Meetings

Throughout the project meetings with stakeholders and supervisors were held to discuss the progress of the project. Meetings were also held internally at the end of each weeks in order to write a weekly status-report for the supervisors. These internal meetings served the purpose of giving the group better insight into how

the project was progressing in terms of what has been completed that week, and where the project is heading in terms of what we wish to do the following week.

Meetings with supervisors were held weekly every Tuesday at 14:30, unless canceled or a different time is agreed upon beforehand.

Bi-weekly meetings with Orange Business were held to update the groups contact within Orange of project progress, these meetings were also used to discuss the current solution to the task with other people within Orange that were interested in the project.

6.4.2 Tools

When planning the project, the group decided on a set of tools to aid the development process of this project:

Communication

The group primarily used two communication platforms: Microsoft Teams [74] and Signal [75]. The message application Signal was used for internal text based communication. If the need arose for voice communication within the group the online platform Discord [76] was used. Microsoft Teams was used for both text based communication with supervisors and meetings. For communications with stakeholder, the group used e-mail.

Collaboration platforms

The group used the online LaTeX writing tool Overleaf [77] for both thesis writing, meeting minutes and status reports. OneDrive [78] was utilized as a backup of all files concerning the project, except code related files. The project source code was kept and versioned in a code repo on Github [79]. As Github also provides a responsive and user-friendly issue board together with its ability to easily integrate commits into issues GitHub rather than Jira [80] was chosen as the groups Kanban board.

Time management

The group recorded hours with the online tool Clockify [81]. This allowed the group to effectively record hours spent on the project, while adding labels and comments to the registered hours. To create a Gantt chart the team used TeamGantt as this is an easy to use and accessible online platform made for creating Gantt charts.

6.4.3 Quality assurance

Code review

Before tasks could be set to completed in the kanban board, it would go into review. The team member that did not work on the task had to review the completed work before it could be set to completed. Following this pipeline served not only to ensure both team members were up to date on the work that had been completed, it ensured the code was readable and that it functioned properly.

Commits

When comitting to the GitHub repository there was written sensible commit messages based on the framework conventional commits [82]. Commits were also connected to a task within the Kanban board. This served to make potential errors in the implementation easier to track down.

Backup

Weekly physical backup of project files was taken on an external SSD during the groups internal weekly review, code files were exempt from this.

Chapter 7

Implementation

This chapter will focus on how our anomaly detection approach was implemented. It covers technical aspects of the solution and describes the different technologies and libraries that were used.

In addition to implementing MIURA, a more complete system has been built to complement it. This system has been built using a technologies that are known to be in use at Orange Business, and it is built in such a way that the detection model can be added to their existing systems without requiring any changes to their current infrastructure. The process of implementing MIURA started by first implementing RePAD2 [2] using PyTorch. This implementation of RePAD2 was tested and compared to the evaluation results found in the referenced paper, before it was modified to include the concept of incremental learning, subsequently becoming MIURA.

7.1 Data Handling

The foundation for the components used here was provided from the Kafka-TIG repository, by Amit Singh on Github [83]. All the following components are deployed using docker compose, a tool for defining and running multi-container applications [84], and is hereby referred to collectively as a stack. The most important parts of this stack is Apache Kafka, Telegraf, InfluxDB and Grafana, which acts as the event streamer, intermediary data handler, database and monitoring dashboard, respectively.

7.1.1 Receiving and storing data

Receiving and storing information, specifically time series data and anomalies, is essential for allowing MIURA to function. This is implemented using the event streamer Apache Kafka, which receives the time series data, the database InfluxDB, which stores the data, and the intermediary data handler Telegraf, that

makes sure the received data reaches the database. This system is based around the use of Docker [85], where each component is contained within an isolated container environment. This allows each component to be separately started, stopped or updated as needed. The biggest advantage to this approach is that it works regardless of the platform

Extending this setup with Docker compose allows all containers to be deployed as a multi-container application [84]. This means that all components are configured using a single file, centralizing configuration. If any changes to this configuration is needed later, docker compose will take care to redeploy the affected components.

Another feature of docker compose is that it allows components to communicate on internal networks, ensuring that port numbers only needed for communication between components are not occupied on the host-computer. This feature also provides DHCP and DNS internally for each component, allowing for the use of service-name instead of IP-address. Code listing 7.4 and 7.5 show examples of this, while Code listings 7.1 and 7.2 show examples of how Apache Kafka and InfluxDB are configured.

```
kafka1:
    image: bitnami/kafka:3.3.1
    container_name: kafka1
    ports:
        #- '9093:9093'
        #- '9092:9092'
    environment:
        - KAFKA_CFG_ZOOKEEPER_CONNECT=zookeeper:2181
        - ALLOW_PLAINTEXT_LISTENER=yes
        - KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP=CLIENT:PLAINTEXT,EXTERNAL:PLAINTEXT
        - KAFKA_CFG_LISTENERS=CLIENT://:9092,EXTERNAL://:9093
        - KAFKA_CFG_ADVERTISED_LISTENERS=CLIENT://kafka1:9092,EXTERNAL://localhost:9093,
        - KAFKA_INTER_BROKER_LISTENER_NAME=CLIENT
        - KAFKA_CFG_AUTO_CREATE_TOPICS_ENABLE=true
    depends_on:
        - zookeeper
    networks:
        - kafka
```

Code listing 7.1: Configuration for Apache Kafka in Docker Compose

```
influxdb:
    image: influxdb:2.7.5
    container_name: influxdb
    ports:
        #- "8083:8083"
        - "8086:8086"
        #- "8090:8090"
    depends_on:
        - zookeeper
        - kafka1
    environment: # parameter in variables.env
        DOCKER_INFLUXDB_INIT: X
        DOCKER_INFLUXDB_INIT_USERNAME: X
        DOCKER_INFLUXDB_INIT_PASSWORD: X
        DOCKER_INFLUXDB_INIT_ORG: X
        DOCKER_INFLUXDB_INIT_BUCKET: X
        DOCKER_INFLUXDB_INIT_ADMIN_TOKEN: X
    networks:
      - db
    volumes:
        - ./data/influxdb/:/var/lib/influxdb
```

Code listing 7.2: Configuration for InfluxDB in Docker Compose

Apache Kafka and InfluxDB are configured with their own networks kafka and db, which are not connected to each other. This means that Apache Kafka and InfluxDB are unable to ever communicate directly, but uses Telegraf to handle the data flow between them. Telegraf does not have a separate network, but instead has access to the network of both Apache Kafka and InfluxDB. An example of this can be seen in Code listing 7.3, showing that it is part of both networks, and that it must wait until Apache Kafka and InfluxDB has started running before being deployed.

```
telegraf:
    image: telegraf:1.29.5
    container_name: telegraf
    environment: # configuration in /telegraf/telegraf.conf
        influxdb_token: X # parameter in variables.env
    depends_on:
        - zookeeper
        - influxdb
        - kafka1
    restart: unless-stopped
    networks:
      - kafka
      - db
    volumes:
        - ./conf/telegraf/telegraf.conf:/etc/telegraf/telegraf.conf:ro
```

Code listing 7.3: Configuration for Telegraf in Docker Compose

To move data between the two components, Telegraf uses plugins [86]. Two of

these plugins allows the data transfer to happen. The first is shown in Code listing 7.4 and allows Telegraf to fetch data from Apache Kafka, from a specific topic (used to separate different streams of data from each other) . The other, which is shown in Code listing 7.5, allows Telegraf to send the data it fetched from Apache Kafka to InfluxDB, into a specific bucket (like a separate NoSQL database).

```
[[inputs.kafka_consumer]]
  ## Kafka brokers.
  ## docker-compose internal address of kakfa
  brokers = ["kafka1:9092"]

  ## Topics to consume.
  topics = [ "cpu_util" ] ## Topic for dataset

  ## Data format to consume.
  data_format = "influx"
```

Code listing 7.4: Grafana queries annotations from InfluxDB

```
[[outputs.influxdb_v2]]
  # Configuration for influxdb server to send metrics to
  urls = ["http://influxdb:8086"] ## Docker-Compose internal address
  token = "random_token" ## token name, setting from config not working
  organization = "ORG" ## orga name, setting from config not working
  bucket = "system_state" ## bucket name / db name, setting from config not working

  ## Write timeout (for the InfluxDB client), formatted as a string.
  timeout = "5s"
```

Code listing 7.5: Grafana queries annotations from InfluxDB

7.1.2 Monitoring and Visualization

The monitoring and visualization of time series data is implemented using Grafana which was implemented in the docker compose as the former components. In Grafana, dashboards can be created to visualize the data that is in the database. This works in two parts, the first queries and visualizes the same data that MIURA uses to detect anomalies. This shows the actual pattern of the data in a view that automatically queries the database and updates itself to stay up to date, in real time. An example of such a query is shown in Code listing 7.6.

```
from(bucket: "system_state")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "B3B")
```

Code listing 7.6: Grafana queries dataset from InfluxDB

The second part queries and visualizes the detections that are made by MIURA.

This automatically annotates, on top of the actual data, when an anomaly was detected. This is also automatically updated in real time. Annotations are set based on time and requires this to be mapped manually based on the query that is sent to the database. As seen in Code listing 7.7, this query looks a bit different. How Grafana visualizes the data and annotates anomalies is shown in Figure 7.1.

```
from(bucket: "anomalies")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "miura-b3b-detection-minmax")
  |> map(fn: (r) => ({
    _time: r._time,
    _value: r._value,
    text: "Anomaly: " + string(v: r._value) // Adjust this line based on your data
  }))
```

Code listing 7.7: Grafana queries annotations from InfluxDB

Figure 7.1: From dashboard in Grafana, showing visualization of time series data and detection annotation.

7.2 MIURA

7.2.1 Deep learning Framework

Choosing which programming language and deep learning framework to use is a fundamental part of creating a deep learning model. As discussed by Lee et al. [8], it is hard to choose the framework that suits a model best as there are few or no relevant evaluations available, making it hard to estimate. As discussed further in [8], it is also clear that the choice of deep learning framework significantly impacts the performance of the resulting model. PyTorch, in particular, has shown the most promising results overall for RePAD [8], which is the predecessor of RePAD2.

Since RePAD and RePAD2 are identical in their approach except for a few logical

differences, it was the most comparable and most conclusive answer on which deep learning framework would perform best. Choosing PyTorch also meant that the code of MIURA would be written in Python, which fit well with it having official libraries for Apache Kafka, InfluxDB and scikit, which is needed for implementing the detection model.

7.2.2 LSTM model

The LSTM model serves as the prediction engine of the detection system and is used to determine expected and unexpected behaviour. The LSTM model was implemented using the LSTM module from PyTorch and initially made to work with a recreation of RePAD2, ensuring it functioned as closely to the original as possible [2]. The LSTM model was later developed further to work with MIURA. This approach provided a reliable starting reference comparable to the original, ensuring correct implementation before further development.

Data scaling

While we will not go into detail on how this is implemented, all data to be used for training the LSTM model is scaled beforehand. Scaling of data refers normalization and standardization of data, and ultimately makes the input data into smaller values while keeping to or close to the original pattern. Exactly how this happens differs between the different possible methods of standardization. Due to how LSTM models function, data is standardized to help the LSTM model better and more efficiently learn the data it is trained on [87]. Data scaling was implemented using the sklearn library from scikit [88] [89].

Training

In order for the LSTM model to accurately predict the next data point, it needs to be trained. This training gives the LSTM model a general idea of the pattern in the data. As described in Section 5.1.4, the LSTM model is given the three most recent data points to train on. These three data point makes up a batch, and comes in pairs of x-values and y-values, where x-values are a constant 1, 2 and 3, and y-values are the data point values. The LSTM model training works by first giving the batch of x-values to the model, then comparing the predicted y-values for each x-value to the actual y-values. The model then tries to slowly correct itself based on the accuracy of those predictions. This is implemented as shown in Code listing 7.8.

```python
def forward(self, x):
    x, _ = self.lstm(x)  # LSTM output
    x = self.fc(x[:, -1, :])  # Extract last output from timstep for prediction
    return x

def train_model(self, x, y):
    optimizer = optim.Adam(self.parameters(), lr=self.learning_rate)
    loss_fn = nn.MSELoss()  # Mean Squared Error Loss

    for _ in range(self.num_epochs):
        self.train()  # Set the model to training mode
        output = self(x)  # Forward pass
        optimizer.zero_grad()  # Clear the gradients
        loss = loss_fn(output, y.view(-1, 1))  # Compute the loss
        loss.backward()  # Backward pass
        optimizer.step()  # Update the weights
```

Code listing 7.8: Training the LSTM model

Prediction

After training, the LSTM model becomes static. This means that some value of X always returns the same value of Y. Note that the same values of X are used every time the LSTM model is trained, and the reason for this is to train the model to accurately predict within the general pattern of the data instead of specific values of X. This all means that when we need the LSTM model to predict the value of the next data point, we always want to predict the value of Y when X is 4. This may lead the LSTM model to be less accurate at predicting specific data points, but it does allow the LSTM model to more accurately give predictions within the general data pattern.

```python
def predict_next(self):
    # Get a prediction
    x = torch.tensor([4], dtype=torch.float32).view(-1, 1, 1)
    x = self.transform_x(x)
    y_pred = self(x)
    y_pred = self.inverse_transform_y(y_pred)
    return y_pred.item()
```

Code listing 7.9: Predicting the next value

7.2.3 Access Data

MIURA queries the existing data points stored as measurements in the database and dynamically adjusts the starting period to avoid querying the same data point twice. This part communicates with the database using the Python client library provided by InfluxDB [90]. It is designed to allow the model to begin from any previous point in time, rather than being limited to only the most recently received data point.

All available data are then queried and received from the database in a single batch, referred to as events, and then processed from oldest and up until there are no more available events left. When the current list of events are all processed, the start time from which data is wanted from the database get dynamically updated to the timestamp of the last event, incremented enough to not also include that last event. The model then waits, polling the database on a regular interval until a new event is available.

```
while True:

  # Construct the Flux query
  query = f'''
  from(bucket: "{bucket}")
   |> range(start: time(v: "{start_time}"))
   |> filter(fn: (r) => r["_measurement"] == "{measurement}")
  '''

  # Query the data
  events = list(query_api.query_stream(org=org, query=query))

  if len(events) > 1: # Need at least 4 to predict next and compar

      for i in range(len(events)-1):
          batch_events.append(events[i])
          next_event = events[i+1]

          .
          .
          .

          # Increment T
          T += 1
      # Update start time for the next iteration
      last_event_time = batch_events[-1].get_time()
      # Increment by 1 second to avoid duplicate events
      start_time = (last_event_time + timedelta(seconds=time_increment)).isoformat()

  else:
      print("No events found in range.")

  time.sleep(poll_interval)
```

Code listing 7.10: Implementation for accessing and querying from the database

7.2.4 Detect Anomalies

The overall functioning of how the detection model detects anomalies was covered earlier in Code listing 5.1 in Chapter 5, in the form of pseudo-code and will not be covered in detail here. Though referring to the beginning of the detection algorithm as seen in Code listing 7.11 a model is first initialized and then further updated, used to predict the next number in line and then stored to later be used for calculating AARE and the threshold, as shown in Equations (5.1) and (5.2).

This part exemplifies how models are trained, how the model is used to produce a prediction, and how the prediction and the actual value is saved to later be used for calculating AARE-values and the new threshold. The predicted values and the actual values are kept in a data structure that automatically retains a size of three, removing the oldest values every time a fourth value is added to it.

```
if T >= 2 and T < 5: # 3 total values when T = 2
    if T == 2:
        # Initialize the model
        M = initialize([event.get_value() for event in list(batch_events)])
    else:
        M = update_model(M, [event.get_value() for event in list(batch_events)])
    pred_D_T_plus_1 = M.predict_next()
    # Append the event and its prediction to the sliding window
    actual_value.append(next_event.get_value())
    predicted_value.append(pred_D_T_plus_1)
```

Code listing 7.11: First instance of initializing and updating an LSTM model in the detector

7.2.5 Reporting anomalies

InfluxDB is different from typical SQL and NoSQL databases in that it stores data as immutable events, meaning that the events cannot be updated. This meant that changing a single field within a data point to mark it as anomalous was not possible. Since InfluxDB stores each set of time series as a measurement, another measurement can be made to keep record of anomalous data points. Storing anomalies this way means keeping duplicated values of the original data point, with the only difference being that they have been marked as anomalous. For every time series measurement there will therefore be kept a complementing measurement containing all reported anomalies.

```
def report_anomaly(T, timestamp, actual_value, predicted_value, write_api):
    point = Point("miura-dataset_name-detection")\
        .tag("host", "detector")\
        .field("T", float(T))\
        .field("actual_value", float(actual_value))\
        .field("predicted_value", float(predicted_value))\
        .time(timestamp, WritePrecision.NS)

    write_api.write(bucket="anomalies", org="ORG", record=point)
```

Code listing 7.12: Reporting anomalies to the database

7.2.6 Difference between MIURA and RePAD2

The main difference between MIURA and RePAD2 is that RePAD2 trains a new LSTM model each time re-training is initiated. During re-training, the same function is called, which takes the three most recent data points, converts them to

tensors, creates a new LSTM model, and trains the model on this data, as shown in code listing 7.13. In contrast, MIURA retains and updates the existing LSTM model instead of replacing it.

```
# Function for creating and training model
def train_model(train_events):
    tensor_y = torch.tensor(train_events, dtype=torch.float32).view(-1, 1, 1)
    tensor_x = torch.tensor([1, 2, 3], dtype=torch.float32).view(-1, 1, 1)
    # Create an instance of the LSTM model
    model = LSTM(tensor_x, tensor_y, num_epochs=50, learning_rate=0.005)
    model.train_model() # Train the model

    return model
```

Code listing 7.13: Creating and training a new LSTM model in the detector

When training a model, the function train_model() is used every time. Code listing 7.14 demonstrates the initial step of the RePAD2 algorithm. In contrast, the MIURA algorithm, as shown in Code Listing 7.11, starts by initializing a model and continuously updating it.

```
# Set T to the length of the sliding
    if T >= 2 and T < 5: # 3 total values when T = 2
        M = train_model([event.get_value() for event in list(batch_events)])
        pred_D_T_plus_1 = M.predict_next()
        # Append the event and its prediction to the sliding window
        actual_value.append(next_event.get_value())
        predicted_value.append(pred_D_T_plus_1)
```

Code listing 7.14: Segment of RePAD2 that happens between T = 2 and T = 4, training an LSTM model

Chapter 8

Evaluation and Results

In this chapter, we will assess MIURA with three distinct data scalers and compare their performance against RePAD2 across several real-world, open-source time series datasets. We will evaluate all variants based on detection accuracy, efficiency, and time consumption, discussing which variant excels and the reasons behind its effectiveness.

8.1 Variants To Be Evaluated

In this thesis, we considered the following three scalers:

- **MinMaxScaler** normalizes data such that all values are in the range between 0 and 1 [88].
- **RobustScaler** standardizes data by removing the median and scaling the data according to the interquartile range [91].
- **StandardScaler** standardizes data by scaling it to a standard normal distribution [92].

In order to investigate how these data scalers affect the performance of MIURA, we implemented MIURA in combination with each of these scalers using PyTorch. To provide a fair and comprehensive comparison, we also re-implemented RePAD2 with these three scalers using PyTorch. In total, we evaluated six variants, denoted by:

- MIURA-MinMax
- MIURA-Robust
- MIURA-Standard
- RePAD2-MinMax
- RePAD2-Robust
- RePAD2-Standard

MIURA-MinMax refers to MIURA implemented with MinMaxScaler, MIURA-Robust refers to MIURA implementd with RobustScaler, and so forth.

8.2 Hyperparameter and Parameter Setting

Table 8.1 lists the hyperparameter and parameter settings used for all the six variants during evaluation. These parameters are identical to those used by Lee et al. in [2] to evaluate their implementation of RePAD2. We intend to use the same settings in order to make our evaluation more closely comparable to theirs. An explanation for each hyperparameter and parameter is shown below:

- **Hidden layers** refers to the numbers of hidden layers in the neural network.
- **Hidden units** are the number of neurons within each hidden layer.
- **Epoch** is the number of training loops used each time an LSTM model is trained.
- **Learning rate** is the size of the increments used to adjust the gradients in the neural network during each training loop.
- **Activation function** refers to the output function used in the LSTM model, which is the default for LSTM models.
- **Random seed** is the number used to generate pseudo-random numbers, making results reproducible by generating the same random numbers every time.

Table 8.1: Hyperparameter and Parameter settings used for our evaluation

Hyperparameters and parameters	**Value**
Number of hidden layers	1
Number of hidden units	10
Number of epochs	50
Learning rate	0.005
Activation function	tanh
Random seed	140

8.3 Real-World Time Series Datasets

In this thesis, we evaluated each of the six mentioned variants using seven real-world, open-source time series datasets, as shown in Table 8.2.

The first time series data is called rds_cpu_utilization_e47b3b (B3B for short) from the Numenta Anomaly Benchmark [93]. The second time series data is called ec2_cpu_utilization_825cc2 (CC2 for short) from the same benchmark. Both time series consist of 4032 data points collected every five minutes. Both B3B and CC2 contain two point anomalies noted by domain experts. The third time series is ex-

actly the same as CC2. However, we considered the point anomaly as a sequential anomaly since it clearly involves more than one anomalous data point, as shown in the top-most sub-figure in Figure 8.7. This time series data is referred to as CC2-seq.

The fourth time series, C6H6(GT) (shortened to C6H6), originates from the multivariate time series data at the UC Irvine Machine Learning Repository [94]. The fifth time series is called PT08.S1(CO) (shortened to PT08_S1), and the sixth is PT08.S2(NMHC) (shortened to PT08_S2), both from the same repository. The last time series, referred to as TEMP, represents temperature data and is also sourced from the same dataset.

The last four time series data consist of 9375 data points collected every hour from March 10th 2004 to April 4th 2005. These four time series data do not contain anomalies as indicated by domain experts; however, there are missing data points. Given that missing data points might indicate a disruption of the continuity of the system, we considered missing data points as anomalies.

Each missing value in these time series data was assigned a value equal or close to -200. To ease model training and anomaly detection, each data point in these time series was incremented by 300. Missing data points were considered as point anomalies or sequential anomalies, depending on whether they were directly adjacent to another missing data point. With this approach, C6H6, PT08_S1, PT08_S2 and TEMP contains 3 point anomalies and 13 sequential anomalies.

Table 8.2: Details of all time series data used for evaluation

Name	**B3B**	**CC2**	**CC2_seq**	**C6H6**	**PT08_S1**	**PT08_S2**	**TEMP**
Nr of data points	4032	4032	4032	9357	9357	9357	9357
Time interval (min)	5	5	5	60	60	60	60
# of point anomalies	2	2	1	3	3	3	3
# of sequential anomalies	0	0	1	13	13	13	13

8.4 Performance Metrics

In this thesis, we consider the following six performance metrics:

- **Precision** measures the accuracy of a model by comparing the number of true positives to all predicted positives (including false positives). Precision is calculated ($TP/(TP+FP)$). This metric is abbreviated as P hereafter.
- **Recall** represents how sensitive the model is, based on whether it is able to detect all anomalies in a dataset. Recall is calculated with ($TP/(TP+FN)$). This metric is abbreviated as R hereafter.
- **F1-score** represents the culmination of precision and recall, and is calculated with ($2x(PrecisionxRecall)/(Precision+Recall)$). This metric is abbreviated as F1 hereafter.

- **Detection time when training is not required** represents the time required to decide if a data point is anomalous or not when LSTM model retraining is not necessary. This metric is abbreviated as DT-NoTrain hereafter.
- **Detection time when training is required** represents the time required to decide if a data point is anomalous or not when LSTM retraining is required. This metric is abbreviated as DT-Train hereafter.
- **LSTM retraining ratio** represents the ratio of training/retraining the LSTM-model to the total amount of processed data points. This is calculated with *(Number of times LSTM model was trained) / (The total number of data points in the dataset)*. This metric is abbreviated as LSTM-ratio hereafter.

Sehili et al. [95] points out that the methods used for evaluating time series anomaly detection within the field are highly flawed. This led us to test various methods of counting true positives (TP), false positives (FP) and false negatives (FN). During this testing, it was concluded that true positives, false positives and false negative should be counted in a similar manner.

This means specifically that, if any point anomaly occurring at time point T can be detected within a period of time, ranging from time point $T-K$ to time point $T+K$, then this anomaly is deemed correctly detected. On the other hand, if any sequential anomaly occurring between time points A and B can be detected within a detection period from time point $A-K$ to time point B, this anomaly is considered correctly detected. However, if any data point outside of any detection period is detected as anomalous, each of those identified data points is counted as a false positive. Lastly, for any anomaly that is not identified, the total number of data points of that anomaly would be counted as false negatives. This evaluation method was utilized in and suggested by [2], where $K = 7$ for minutely time series, and $K = 3$ for hourly time series based on [96].

8.5 Evaluation Environment

All the evaluations were conducted on a Macbook Pro 15" purchased in 2018. The specific model is equipped with a 6-core Intel Core i7 (i7-8750H) and 16GB of DDR4 SDRAM. The laptop was running MacOS Sonoma 14.4.1. The specific version of Python used was 3.11.7. For PyTorch, version 1.13.1 was employed. Other libraries included numpy, scikit-learn, and influxdb-client, with versions 1.25.4, 1.1.3, and 1.41, respectively.

The detection accuracy for all variants and time series data was assessed using automated scripts that followed the exact method described in Section 8.4. These scripts are available in the accompanying ZIP file.

8.6 Evaluation Results

In this section, we present the evaluation results for all six variants across each of the seven time series.

8.6.1 B3B

Table 8.3 provides detailed performance results for all six variants on B3B, while Figure 8.1 illustrates all data points identified as anomalous by these variants, with each detected anomaly marked as '1' in the Y-axis. It is evident that MIURA, when using a scaler, achieves better detection accuracy than RePAD2 using the same scaler. This suggests that incorporating the concept of incremental learning helps enhance detection accuracy, regardless of the scaler used.

Additionally, the results shows that both MIURA and RePAD2 perform the best when utilizing MinMaxScaler, indicating that the choice of the scaler indeed affects the detection performance of MIURA and RePAD2. To explain why MIURA and RePAD2 achieved the best F1-score when utilizing MinMaxScaler, it is evident from Figure 8.2 that these two variants produced more accurate predictions than the others, resulting in lower AARE values. This in turn helped maintain a lower detection threshold, as demonstrated in Figure 8.3.

When it comes to time efficiency, it is evident that all variants are highly efficient, as indicated by their low DT-NoTrain and DT-Train values shown in Table 8.3. Nevertheless, MIURA utilizing a specific scaler demonstrates slightly higher efficiency than RePAD2 utilizing the same scaler.

Table 8.3: Performance results of all variants on the B3B time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	0.786	1	**0.880**	0.0045(=18/4031)	0.0011	0.0024	0.0586	0.0019
MIURA-Robust	0.563	1	0.720	0.0050(=20/4031)	0.0011	0.0023	0.0586	0.0013
MIURA-Standard	0.643	1	0.783	0.0050(=20/4031)	0.0011	0.0027	0.0595	0.0015
RePAD2-MinMax	0.727	1	0.842	0.0037(=15/4031)	0.0011	0.0025	0.0601	0.0018
RePAD2-Robust	0.539	1	0.700	0.0045(=18/4031)	0.0012	0.0027	0.0641	0.0059
RePAD2-Standard	0.539	1	0.700	0.0045(=18/4031)	0.0011	0.0026	0.0600	0.0016

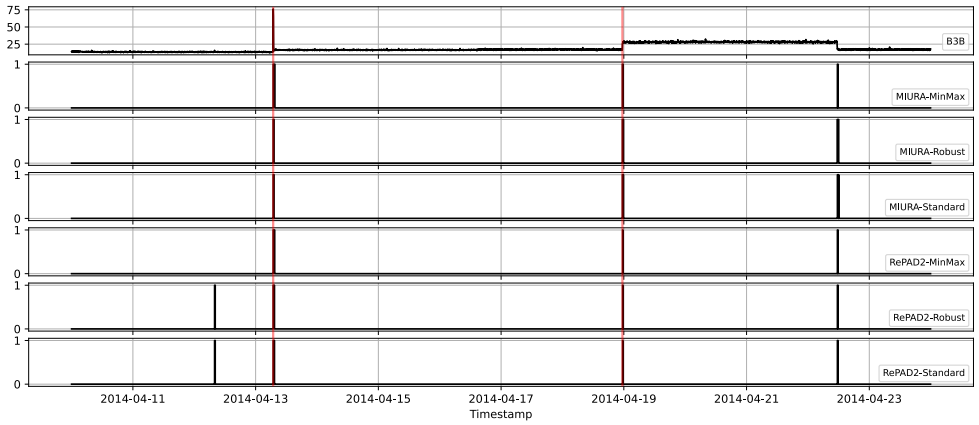
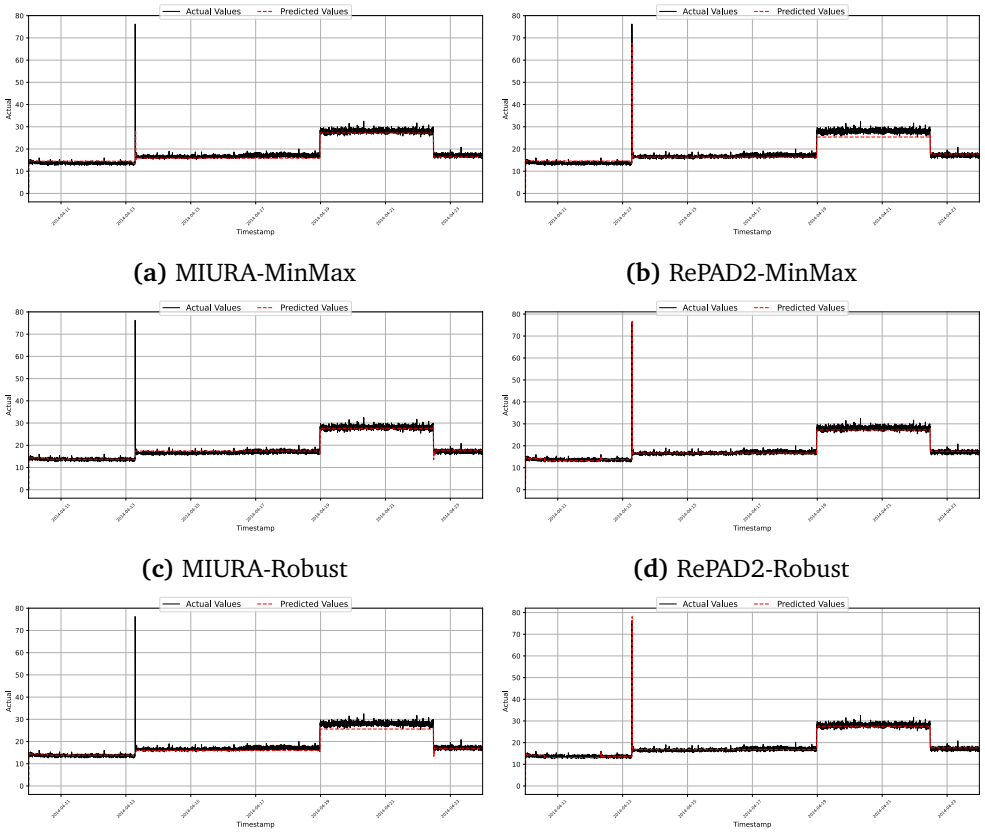


Figure 8.1: The B3B time series vs detection results of all variants: Each anomaly is denoted in red, and each data point detected as anomalous is marked with a '1' in the Y-axis.

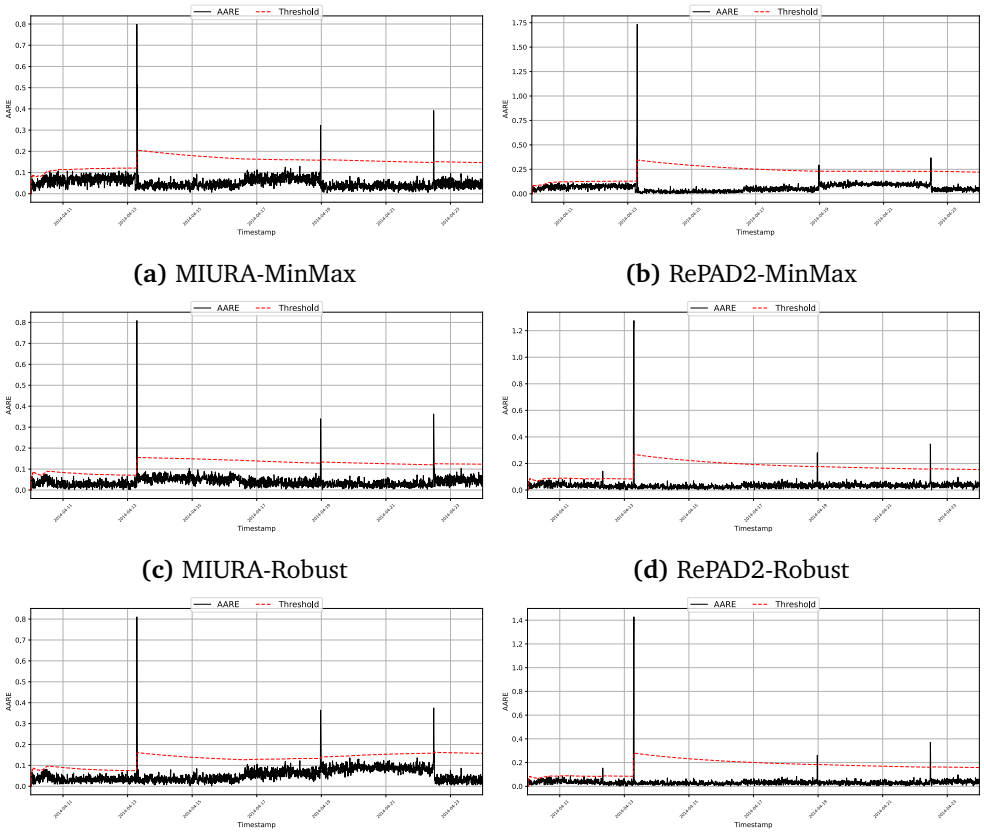


(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.2: All actual values vs predicted values over time for all variants on B3B



(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.3: All derived AARE values vs detection threshold over time for all variants on B3B

8.6.2 CC2

Table 8.4 lists detailed performance results for all the six variants on CC2. These results are similar to the previous time series, and they show that both MIURA and RePAD2 perform the best when utilizing MinMaxScaler.

However, Figure 8.5 demonstrates that the predication produced by MIURA is significantly worse than those of RePAD2, producing higher AARE values, as shown in Figure 8.6. This is because the LSTM model used by MIURA was retained throughout the entire time series. Although the model was updated using the three most recent data points when it failed to predict the next data point, these recent data points often contain significant values. This negatively affects the prediction ability of MIURA as it cannot forget these data points as RePAD2 can. Nevertheless, in this experiment, we can see that MIURA using a specific scaler still performs slightly better than RePAD2 using the same scaler.

As for time efficiency, the DT-NoTrain and DT-Train values are similarly low for all variants, with MIURA having slightly lower DT-Train time. Although MIURA required more retraining than RePAD2, it did not notably increase its processing time. We can see that MIURA is slightly more efficient than RePAD2.

Table 8.4: Performance results of all variants on the CC2 time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	0.429	0.900	**0.581**	0.0062(=25/4031)	0.0011	0.0024	0.0593	0.0027
MIURA-Robust	0.357	1	0.526	0.0089(=36/4031)	0.0012	0.0030	0.0589	0.0012
MIURA-Standard	0.321	0.900	0.474	0.0082(=33/4031)	0.0011	0.0026	0.0591	0.0021
RePAD2-MinMax	0.385	0.833	0.526	0.0042(=17/4031)	0.0011	0.0025	0.0601	0.0020
RePAD2-Robust	0.353	0.857	0.500	0.0055(=22/4031)	0.0012	0.0027	0.0609	0.0011
RePAD2-Standard	0.353	0.851	0.500	0.0055(=22/4031)	0.0012	0.0028	0.0615	0.0024

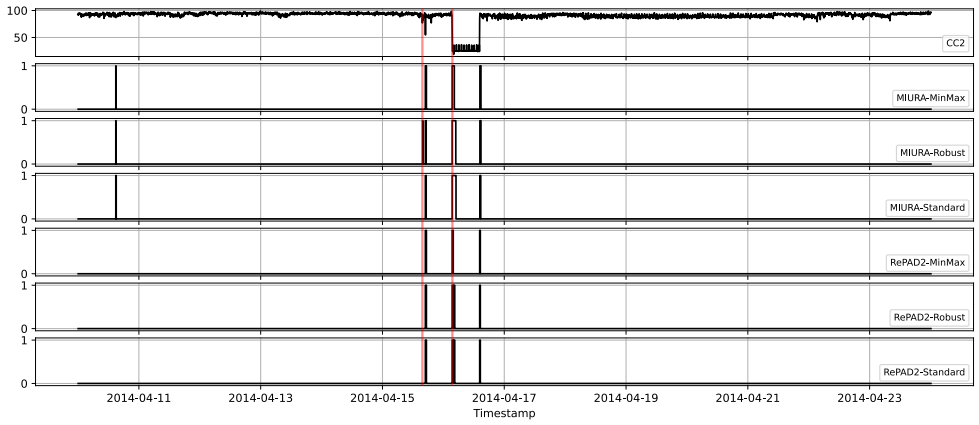
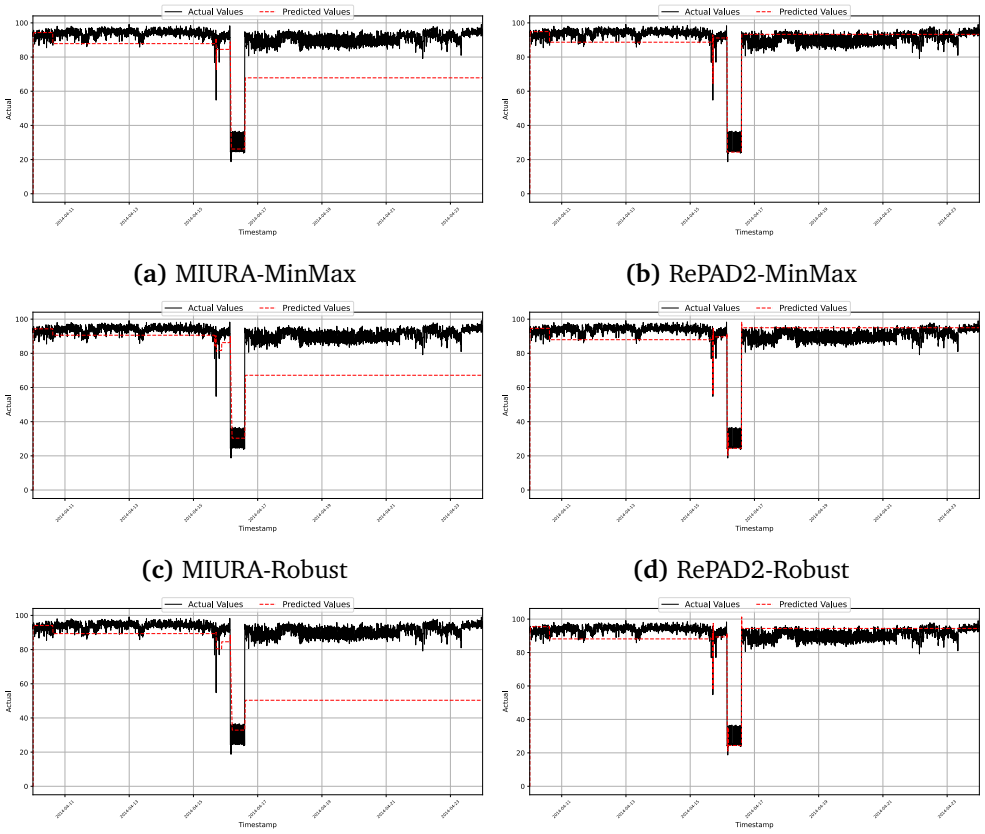


Figure 8.4: The CC2 time series vs detection results of all variants: Each anomaly is denoted in red, and each data point detected as anomalous is marked with a '1' in the Y-axis.

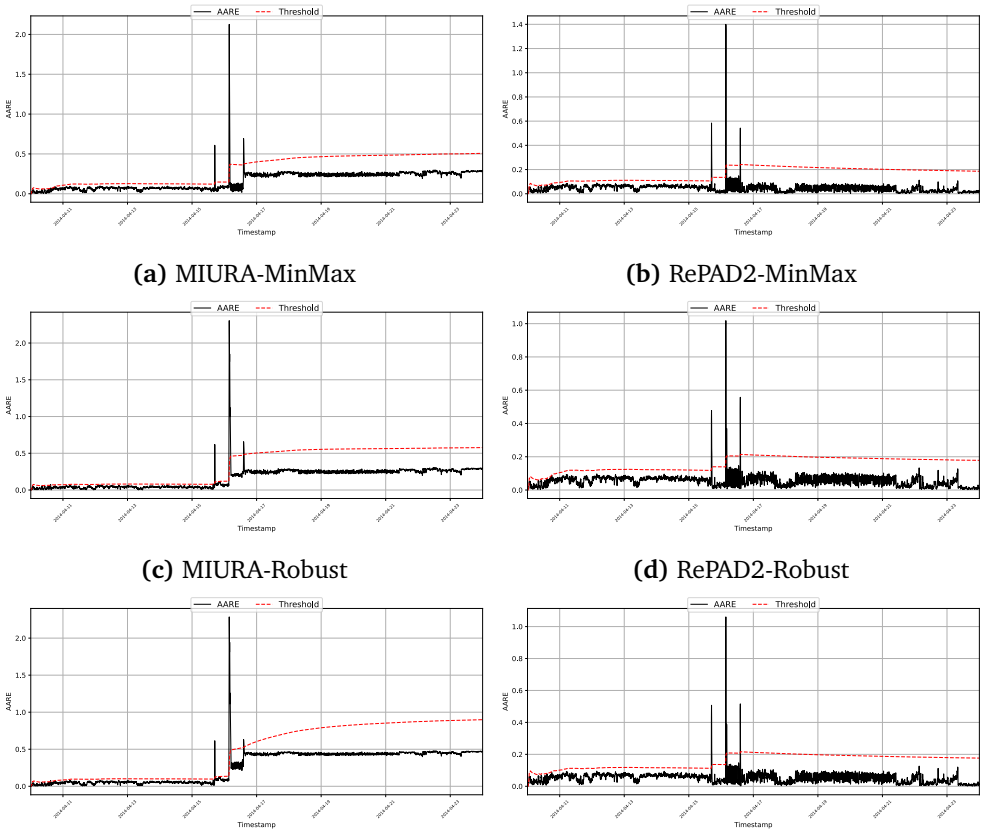


(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.5: All actual values vs predicted values over time for all variants on CC2



(a) MIURA-MinMax

(b) RePAD2-MinMax

(c) MIURA-Robust

(d) RePAD2-Robust

(e) MIURA-Standard

(f) RePAD2-Standard

Figure 8.6: All derived AARE values vs detection threshold over time for all variants on CC2

8.6.3 CC2-seq

In this experiment, all variants were individually applied to CC2-seq. Given that CC2-seq is identical to CC2 except that the point anomaly in CC2 was considered as a sequential anomaly in CC2-seq. Because of this, all variants have the same performance results as shown in Figures 8.5 and 8.6.

As shown in Table 8.5, we observed similar results, i.e., MIURA using a specific scaler outperforms RePAD2 using the same scaler. An examination of the results in Table 8.5 shows that MIURA performed best when utilizing RobustScaler, achieving the highest precision and being the only variant capable of identifying all anomalies.

Table 8.5: Performance results of all variants on the CC2-seq time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	0.714	0.938	0.811	0.0062(=25/4031)	0.0011	0.0024	0.0593	0.0027
MIURA-Robust	0.821	1	**0.902**	0.0089(=36/4031)	0.0012	0.030	0.0589	0.0012
MIURA-Standard	0.786	0.957	0.863	0.0082(=33/4031)	0.0011	0.0026	0.0591	0.0021
RePAD2-MinMax	0.615	0.889	0.727	0.0042(=17/4031)	0.0011	0.0025	0.0601	0.0020
RePAD2-Robust	0.706	0.923	0.800	0.0055(=22/4031)	0.0012	0.0027	0.0609	0.0011
RePAD2-Standard	0.706	0.923	0.800	0.0055(=22/4031)	0.0012	0.0028	0.0615	0.0024

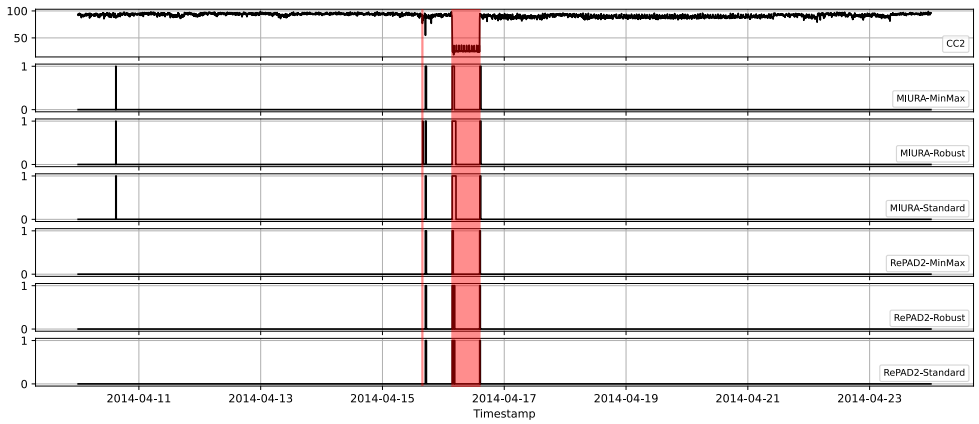


Figure 8.7: Detection result from all variants on CC2-seq

8.6.4 C6H6

When the six variants were applied to detect anomalies in the C6H6 time series, the corresponding detection results differed significantly from those of the previous three time series. As shown in Table 8.6, all RePAD2 variants outperform all MIURA variants. Further examination of Figure 8.8 shows that MIURA was unable to successfully detect most of the anomalies in the C6H6 series, resulting in a low recall value, regardless of the scaler used. This is because, as illustrated in Figure 8.9, MIRUA struggled to accurately predict the time series because it did not disregard the significant values of the first and second anomalies, which negatively affected its prediction capability. In comparison, RePAD2 provided better predictions, which significantly affected the corresponding AARE values and detection thresholds as shown in Figure 8.10. This enables more effective anomaly detection compared to MIURA.

It is also interesting to note that all MIURA variants exhibited a lower LSTM retraining ratio compared to the three RePAD2 variants (see Table 8.6). This is because the three MIURA variants had higher detection thresholds relative to their corresponding AARE values as illustrated in Figure 8.10. This phenomenon results in less frequent model retraining compared to the three RePAD2 variants.

Furthermore, all variants display similarly low DT-NoTrain values, indicating their capability for real-time anomaly detection. However, when LSTM model retraining is required, MIURA does not maintain the same level of efficiency observed in the previous three time series. Specifically, when paired with a particular scaler, MIURA requires a comparable amount of time to RePAD2 using the same scaler.

Table 8.6: Performance results of all variants on the C6H6 time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	0.957	0.061	0.115	0.0029(=27/9356)	0.0017	0.0019	0.0677	0.0011
MIURA-Robust	0.977	0.144	0.251	0.0052(=49/9356)	0.0016	0.0021	0.0606	0.0023
MIURA-Standard	0.978	0.136	0.238	0.0056(=52/9356)	0.0015	0.0018	0.0624	0.0046
RePAD2-MinMax	0.986	1	**0.993**	0.0101(=95/9356)	0.0016	0.0034	0.0601	0.0014
RePAD2-Robust	0.983	1	0.991	0.0092(=86/9356)	0.0017	0.0035	0.0617	0.0031
RePAD2-Standard	0.983	1	0.991	0.0092(=85/9356)	0.0017	0.0036	0.0625	0.0029

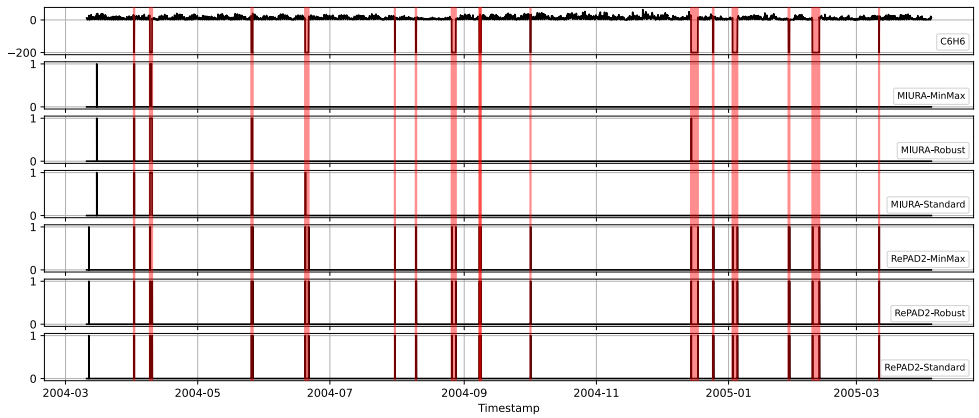
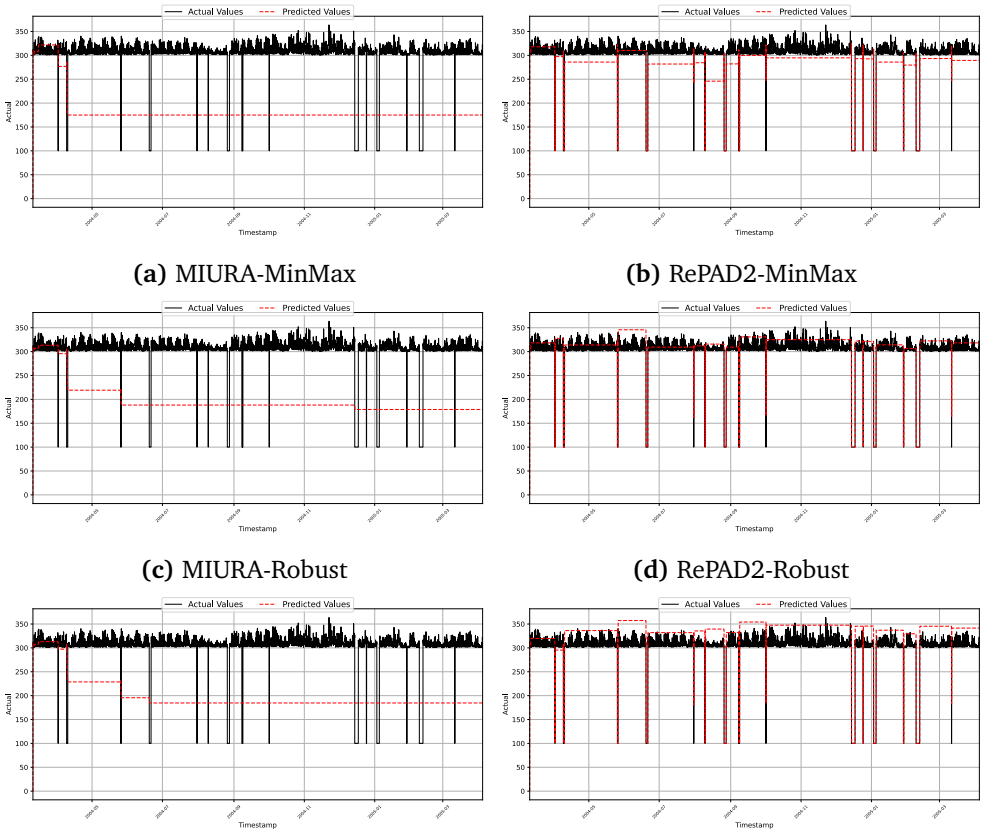


Figure 8.8: The C6H6 time series vs detection results of all variants: Each anomaly is denoted in red, and each data point detected as anomalous is marked with a '1' in the Y-axis.

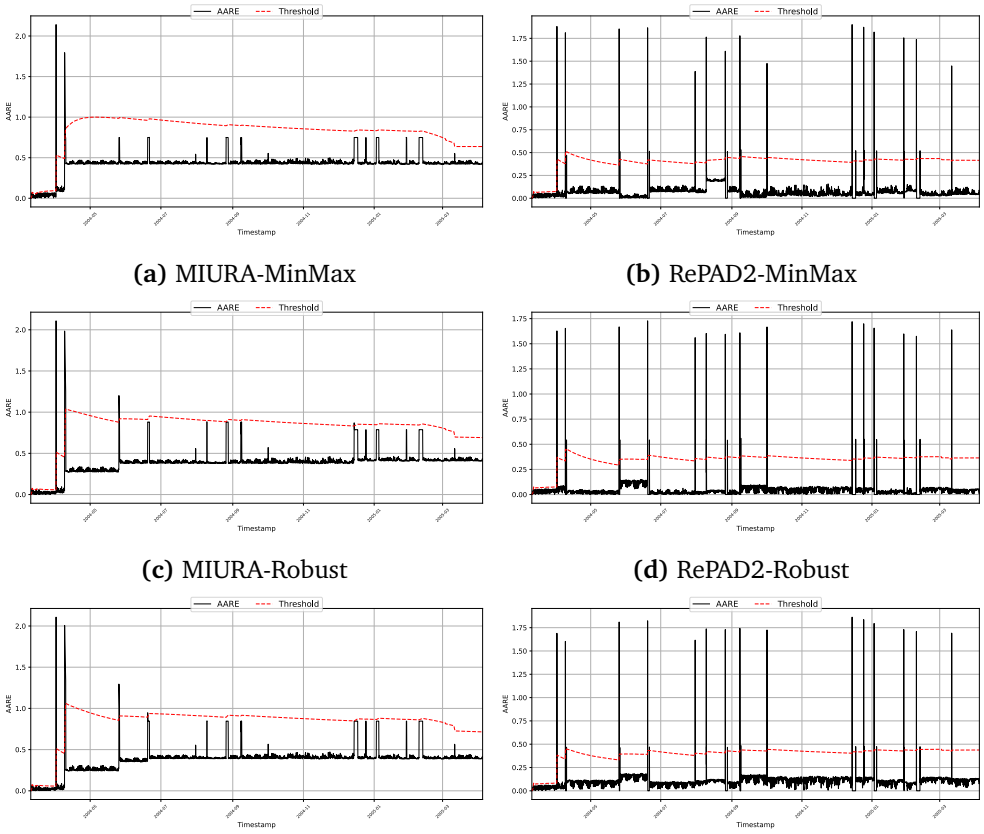


(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.9: All actual values vs predicted values over time for all variants on C6H6



(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.10: All derived AARE values vs detection threshold over time for all variants on C6H6

8.6.5 PT08_S1

Table 8.7 provides detailed performance results for all six variants on PT08_S1, while Figure 8.11 illustrates all data points identified as anomalous by these variants. This time series posed a challenge to both MIURA and RePAD2, as all variants exhibited extremely low recall. Although all variants achieved very high precision, with most nearing perfection, the recall was so low that only one variant achieved an F1-score higher than 0.1. The reason for these outcomes is evident in Figure 8.12; all variants were failed to accurately predict the PT08_S1 time series, leading to to ineffective AARE values and detection thresholds. If we examine the PT08_S1 time series more closely, we can see that it contains higher fluctuations compared to all the previous time series. The results suggest that both MIURA and RePAD2 failed to handle this kind of time series, even with the incorporation of incremental learning.

In terms of time efficiency, all variants triggered only a few model retraining, as indicated in Figure 8.13. They had similar results for DT-NoTrain, while MIRUA was slightly more efficient in terms of DT-Train.

Table 8.7: Performance results of all variants on the PT08_S1 time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	1	0.026	0.050	0.0012(=11/9356)	0.0016	0.0016	0.0605	0.0024
MIURA-Robust	1	0.034	0.066	0.0015(=14/9356)	0.0016	0.0014	0.0588	0.0010
MIURA-Standard	0.941	0.057	**0.108**	0.0022(=21/9356)	0.0015	0.0016	0.0598	0.0018
RePAD2-MinMax	1	0.023	0.045	0.0011(=10/9356)	0.0015	0.0015	0.0629	0.0032
RePAD2-Robust	1	0.008	0.016	0.0006(=6/9356)	0.0015	0.0015	0.0606	0.0019
RePAD2-Standard	0.875	0.020	0.040	0.0013(=12/9356)	0.0015	0.0017	0.0608	0.0011

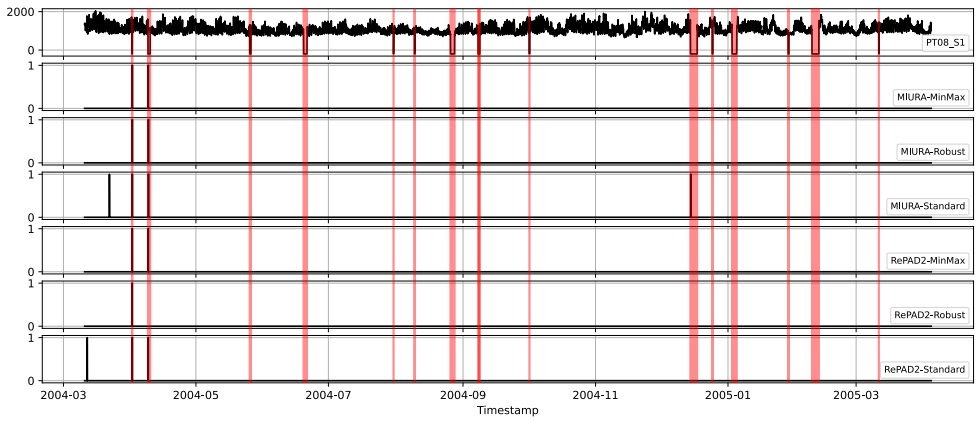
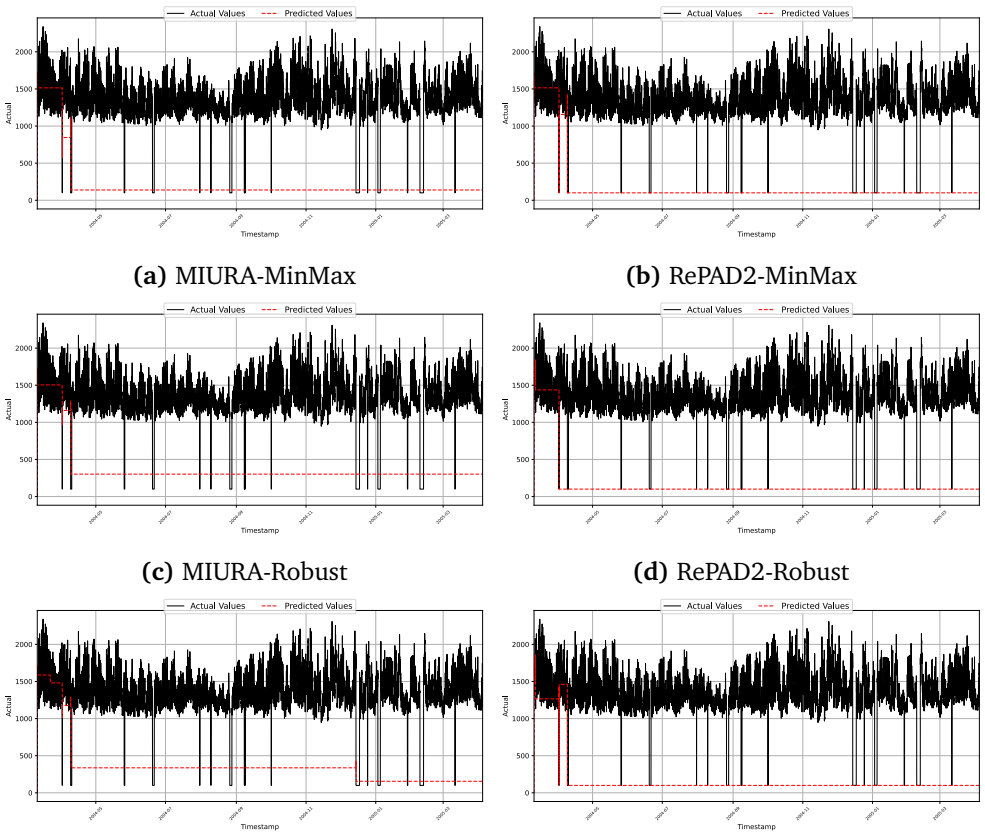


Figure 8.11: The PT08_S1 time series vs detection results of all variants: Each anomaly is denoted in red, and each data point detected as anomalous is marked with a '1' in the Y-axis.



(a) MIURA-MinMax

(b) RePAD2-MinMax

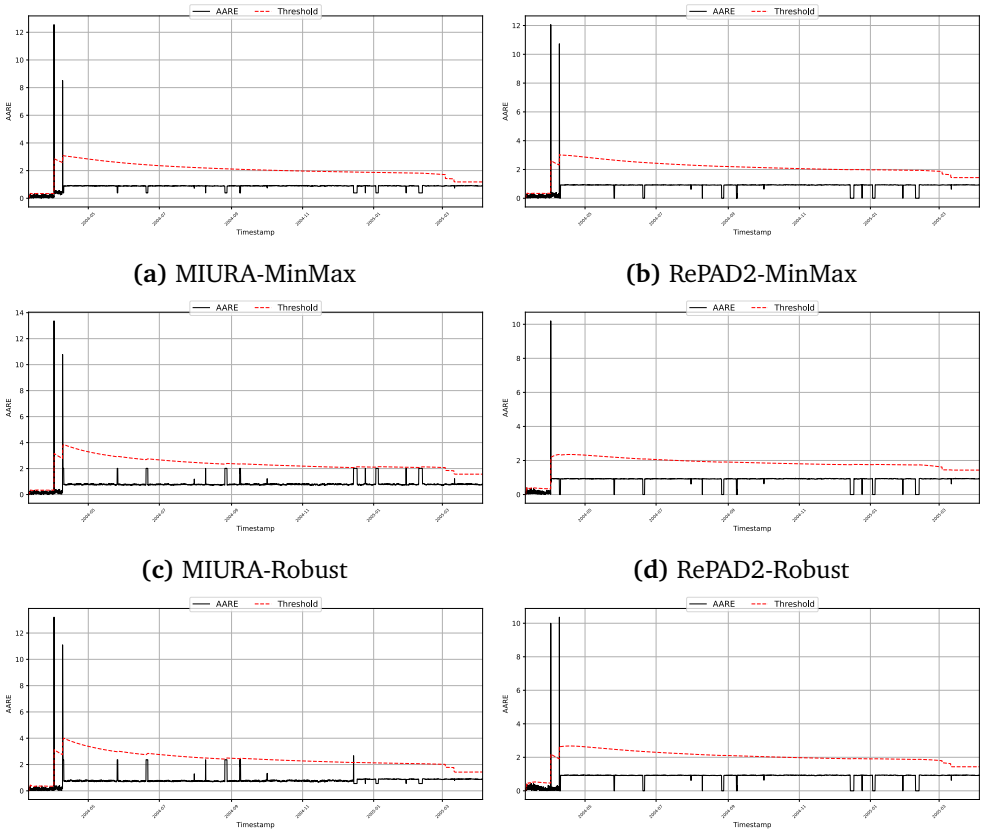
(c) MIURA-Robust

(d) RePAD2-Robust

(e) MIURA-Standard

(f) RePAD2-Standard

Figure 8.12: All actual values vs predicted values over time for all variants on PT08_S1



(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.13: All derived AARE values vs detection threshold over time for all variants on PT08_S1

8.6.6 PT08_S2

When all variants were applied to the PT08_S2 time series, their performance was almost identical to that on the PT08_S1 time series, as discussed in Section 8.6.5.

As shown in Table 8.8, all variants had extremely low recall values, even though they each achieved a precision of 1. Figure 8.14 further illustrates that many anomalies could not be detected by each of these variants. By examining their prediction results shown in Figure 8.15, it is clear why they had poor detection accuracy, as they all failed to accurately predict the PT08_S2 time series. As a result, their detection thresholds are completely useless for determining the anomalies in the PT08_S2 time series (see Figure 8.16).

Table 8.8: Performance results of all variants on the PT08_S2 time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	1	0.020	0.040	0.0010(=9/9356)	0.0015	0.0014	0.0591	0.0040
MIURA-Robust	1	0.029	0.056	0.0013(=12/9356)	0.0016	0.0014	0.0600	0.0023
MIURA-Standard	1	0.031	**0.061**	0.0014(=13/9356)	0.0015	0.0014	0.0606	0.0020
RePAD2-MinMax	1	0.023	0.045	0.0011(=10/9356)	0.0017	0.0017	0.0778	0.0051
RePAD2-Robust	1	0.0082	0.016	0.0004(=4/9356)	0.0015	0.0014	0.0651	0.0074
RePAD2-Standard	1	0.020	0.040	0.0010(=9/9356)	0.0017	0.0023	0.0631	0.0029

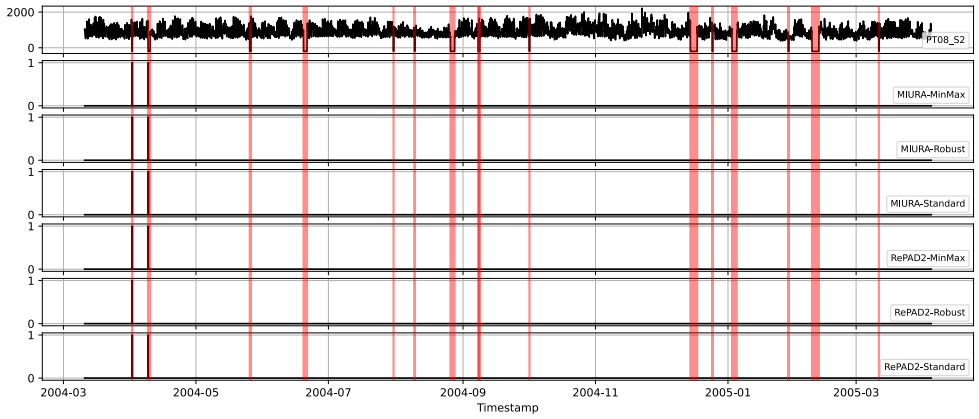
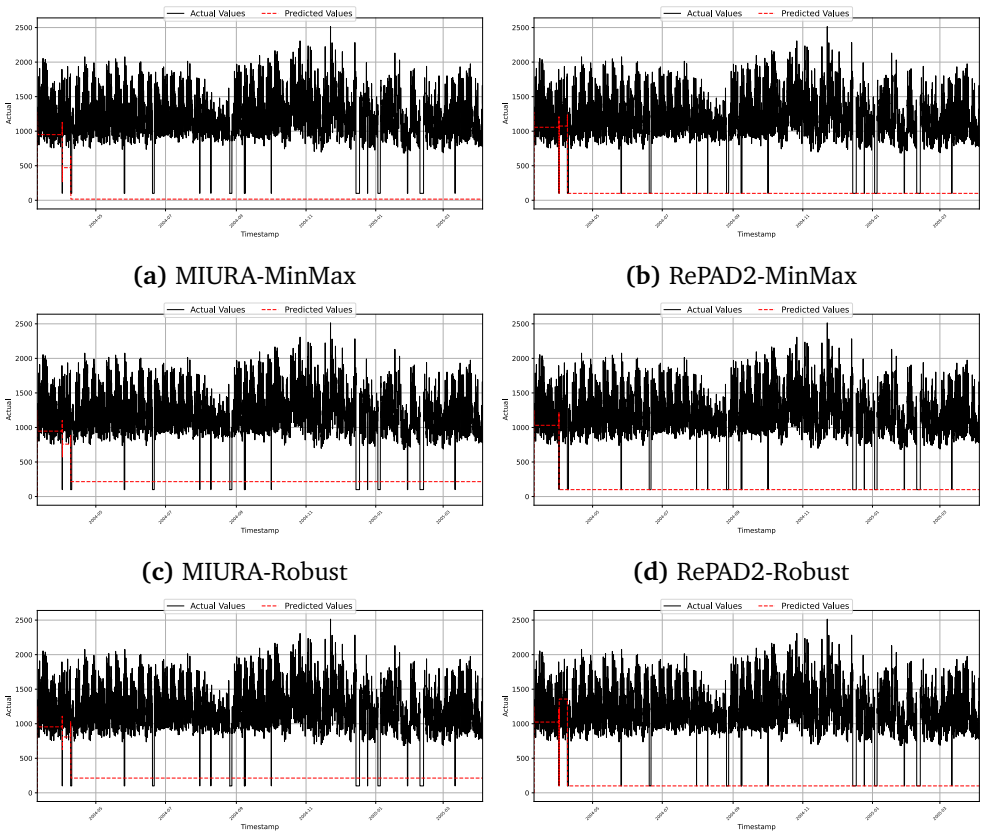


Figure 8.14: The PT08_S2 time series vs detection results of all variants: Each anomaly is denoted in red, and each data point detected as anomalous is marked with a '1' in the Y-axis.

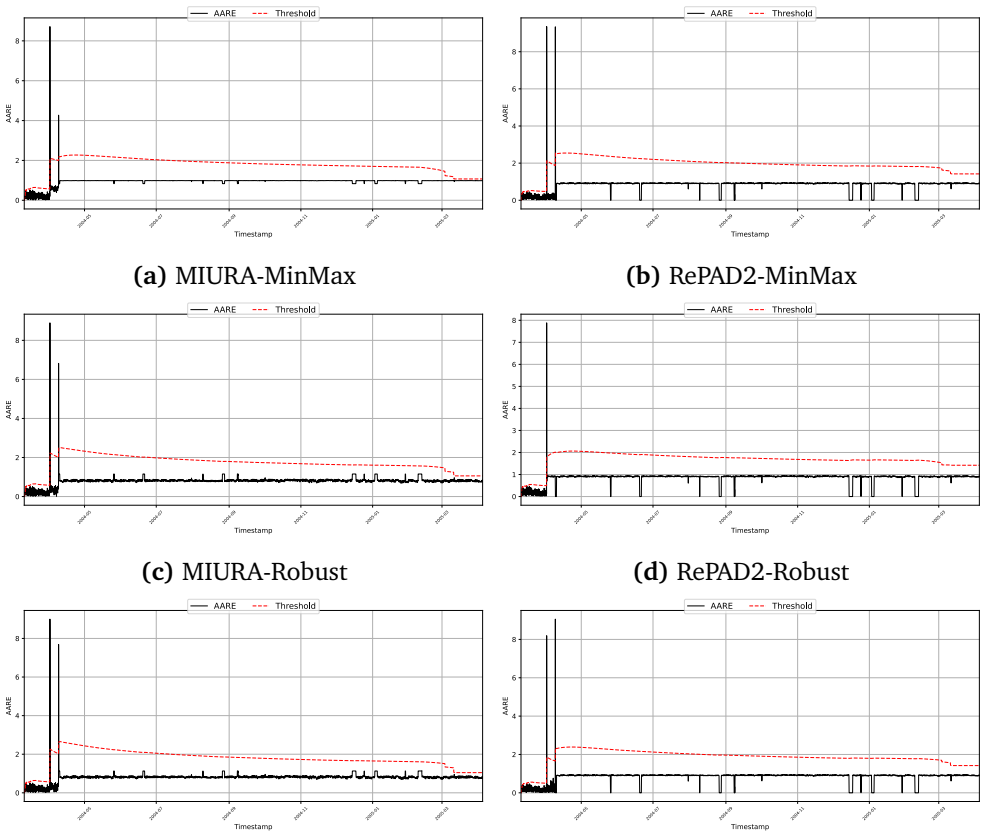


(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.15: All actual values vs predicted values over time for all variants on PT08_S2



(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.16: All derived AARE values vs detection threshold over time for all variants on PT08_S2

8.6.7 TEMP

When the six variants were applied to the TEMP time series, they exhibited similar detection results as those observed with the C6H6 time series. All variants achieved a precision of 1 or close to 1 (see Table 8.9), but MIURA failed to identify more than half of the total anomalies, as shown in Figure 8.17. On the other hand, all RePAD2 variants identified all anomalies except one, leading to a much higher F1-score than MIURA. The main reason behind this is their prediction results. It is clear that MIURA was unable to accurately predict the TEMP time series due to its incremental learning design, leading to ineffective detection thresholds (see Figures 8.18 and 8.19).

In addition, similar to the results for the other six time series, both MIURA and RePAD2 demonstrated their capability for real-time anomaly detection, regardless of whether LSTM model retraining was required.

Table 8.9: Performance results of all variants on the TEMP time series

	Accuracy			**LSTM retraining ratio**	**DT-NoTrain (sec)**		**DT-Train (sec)**	
	P	R	F1		Avg.	Std.	Avg.	Std.
MIURA-MinMax	0.973	0.120	0.214	0.0048(=45/9356)	0.0016	0.0021	0.0590	0.0011
MIURA-Robust	0.972	0.237	0.382	0.0087(=81/9356)	0.0017	0.0025	0.0661	0.0219
MIURA-Standard	0.975	0.260	0.410	0.0094(=88/9356)	0.0015	0.0020	0.0689	0.0427
RePAD2-MinMax	1	0.761	0.865	0.0106(=99/9356)	0.0017	0.0037	0.0609	0.0014
RePAD2-Robust	1	0.814	**0.898**	0.0091(=85/9356)	0.0017	0.0035	0.0616	0.0025
RePAD2-Standard	1	0.814	**0.898**	0.0090(=84/9356)	0.0016	0.0035	0.0625	0.0028

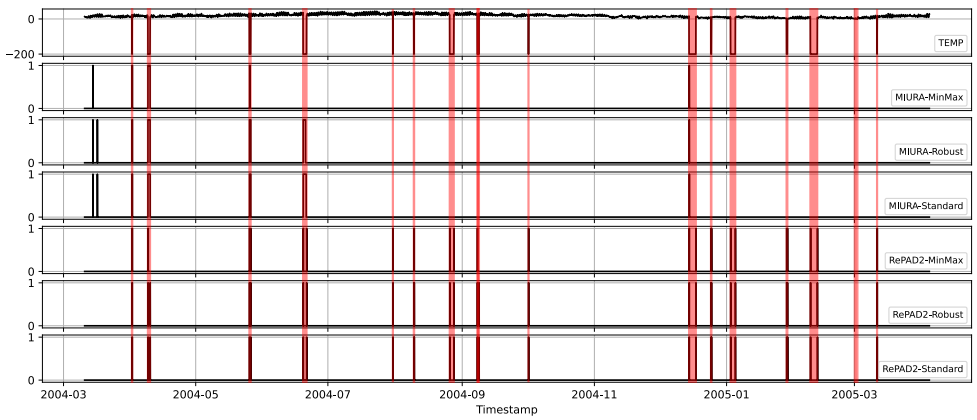
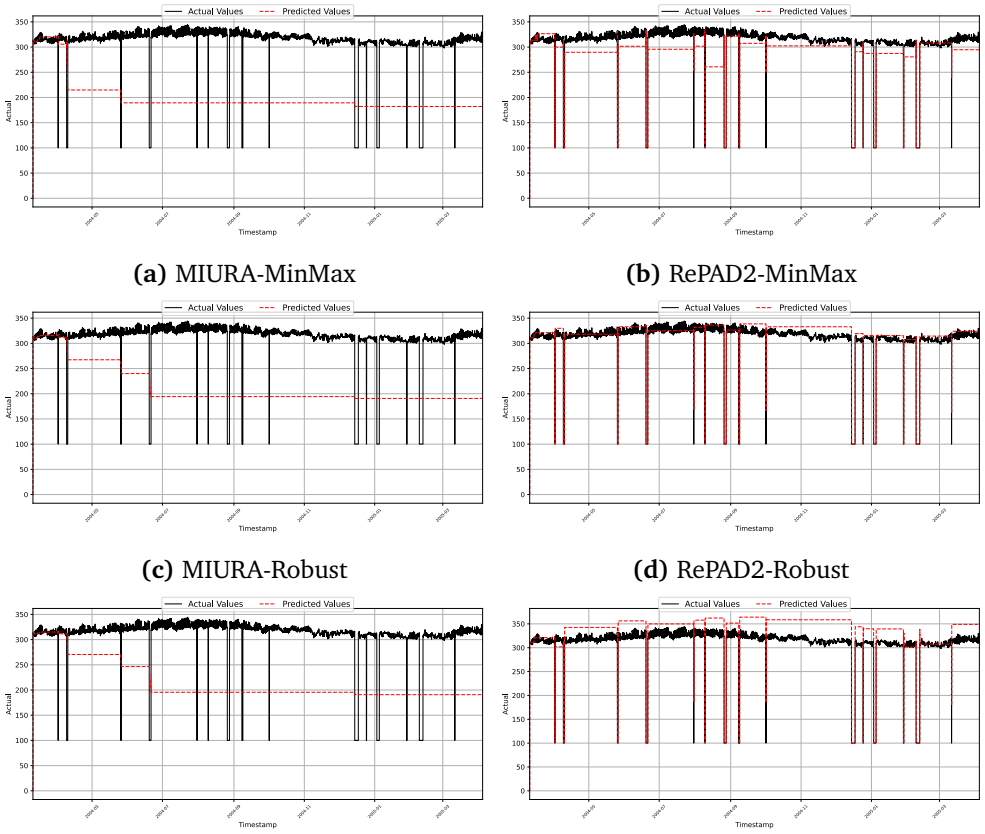


Figure 8.17: The TEMP time series vs detection results of all variants: Each anomaly is denoted in red, and each data point detected as anomalous is marked with a '1' in the Y-axis.



(a) MIURA-MinMax

(b) RePAD2-MinMax

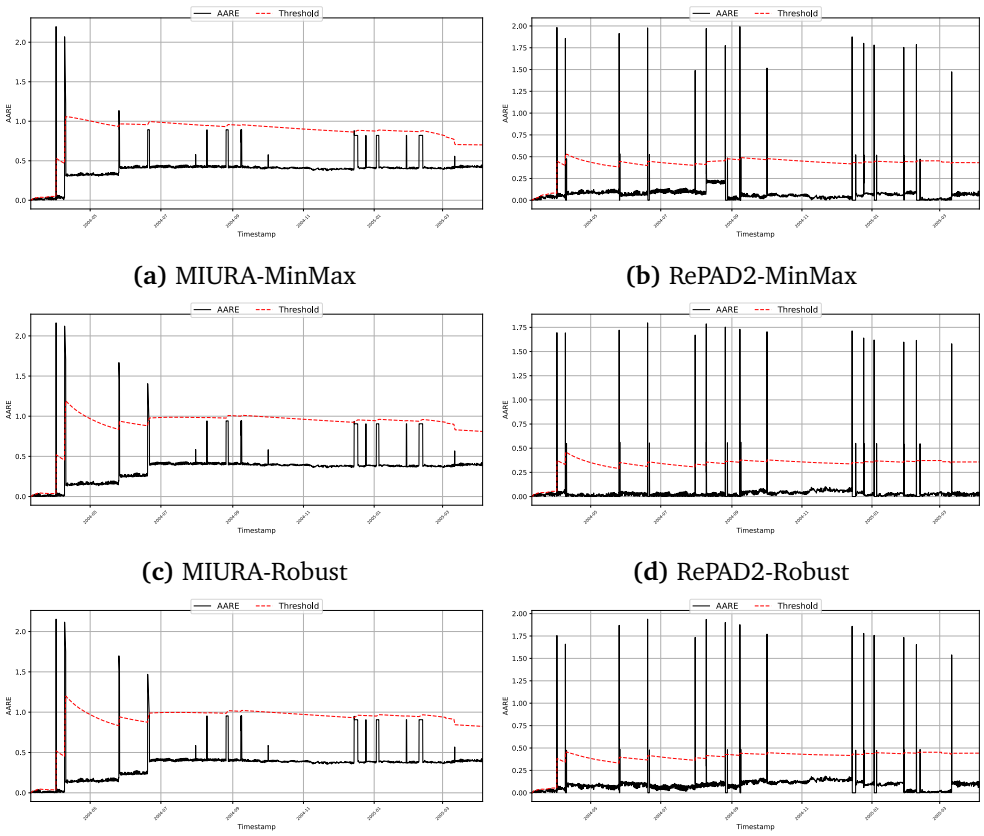
(c) MIURA-Robust

(d) RePAD2-Robust

(e) MIURA-Standard

(f) RePAD2-Standard

Figure 8.18: All actual values vs predicted values over time for all variants on TEMP



(a) MIURA-MinMax **(b)** RePAD2-MinMax

(c) MIURA-Robust **(d)** RePAD2-Robust

(e) MIURA-Standard **(f)** RePAD2-Standard

Figure 8.19: All derived AARE values vs detection threshold over time for all variants on TEMP

8.7 Summary

Table 8.10 summaries the detection accuracy of all variants across the seven time series. According to all previous evaluation results, MIURA provides better detection accuracy than RePAD2 when applied to time series with minor fluctuations, such as B3B and CC2. When applied to time series with higher fluctuations, such as C6H6 and TEMP, RePAD2 offers better detection accuracy. However, for time series with very high fluctuations, such as PT08_S1 and PT08_S2, both methods fail to effectively detect anomalies. Therefore, if users are aware of the characteristics of their time series, they can choose either MIURA or RePAD2, or explore other strategies such as increasing the number of data points for model training or incorporating methods for detecting recurrent patterns.

In addition, the choice of scalers makes a difference, as different scalers impact the detection performance of both MIURA and RePAD2 as observed in the previous subsections. If we exclude MIURA's results on PT08_S1 and PT08_S2, it is

recommended to use MinMaxScaler or RobustScaler for MIURA because they led to better detection accuracy.

Finally, all the evaluation results demonstrate that both MIURA and RePAD2 are highly efficient in determining whether a data point is anomalous. It is also clear that MIURA in general provides slightly better time efficiency than RePAD2, especially when LSTM model retraining is required. In other words, incorporating incremental learning indeed helps enhance the efficiency of anomaly detection.

Table 8.10: Detection accuracy of all variants across different time series.

	B3B			**CC2**			**CC2_seq**			**C6H6**			**PT08_S1**			**PT08_2**			**TEMP**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
MIURA-MinMax	0.786	1	**0.880**	0.429	0.900	**0.581**	0.714	0.938	0.811	0.957	0.061	0.115	1	0.026	0.050	1	0.020	0.040	0.973	0.120	0.214
MIURA-Robust	0.563	1	0.720	0.357	1	0.526	0.821	1	**0.902**	0.977	0.144	0.251	1	0.034	0.066	1	0.029	0.056	0.0972	0.237	0.382
MIURA-Standard	0.643	1	0.783	0.321	0.900	0.474	0.786	0.957	0.863	0.978	0.136	0.238	0.941	0.057	**0.108**	1	0.031	**0.061**	0.975	0.260	0.410
RePAD2-MinMax	0.727	1	0.842	0.345	0.833	0.526	0.615	0.889	0.727	0.986	1	**0.993**	1	0.023	0.045	1	0.023	0.045	1	0.761	0.865
RePAD2-Robust	0.539	1	0.700	0.353	0.857	0.500	0.706	0.923	0.800	0.983	1	0.991	1	0.008	0.016	1	0.008	0.016	1	0.814	**0.898**
RePAD2-Standard	0.539	1	0.700	0.353	0.857	0.500	0.706	0.923	0.800	0.983	1	0.991	0.875	0.020	0.040	1	0.020	0.040	1	0.814	**0.898**

Chapter 9

Evaluation of Requirements

This chapter will discuss if our solution and implementation successfully satisfy the requirements set in Chapter 4.

9.1 Functional Requirements

Requirement F1: Store data persistently

Persistent storage was achieved through the implementation of the time series database InfluxDB. The database stores both received time series data and anomalies reported by MIURA. This is described further in Section 5.1.3.

Requirement F2: Receive time series data in real-time

This process was emulated using a Python-script, acting as the data source. Time series data is sent to Apache Kafka, and is then transferred to InfluxDB by Telegraf. All data is available from InfluxDB within seconds if it being sent, ensuring real-time transfer and availability of time series data.

Requirement F3: Access and process received data

MIURA is able to query time series data directly from InfluxDB, fetching all new data as a single batch. MIURA then processes all received data before again querying InfluxDB until new data is available. This makes it possible for MIURA to begin detection from any previous point in time.

Requirement F4: MIURA must be able to incrementally train an LSTM-model

Rather than always retraining a new LSTM model, like RePAD2, MIURA is able to keep training a LSTM model continuously, and was shown to increase detec-

tion accuracy on multiple time series data compared to RePAD2. How this was implemented is detailed in Chapter 5.

Requirement F5: MIURA must be able to report anomalies when detected

MIURA is able to report detected anomalies by communicating directly with InfluxDB, reporting anomalies continuously as they identified. This is further described in Section 7.2.5.

Requirement F6: Visual monitoring of detected anomalies in real-time

Anomalies are reported directly to InfluxDB by MIURA, which then makes the data available to Grafana. Both normal data and reported anomalies are then queried by Grafana and visualized in real time. This is detailed in Section 7.1.2.

9.2 Non-Functional Requirements

Requirement NF1: MIURA must be able to process data faster than it arrives

It was shown during the evaluation of MIURA that the approach is highly efficient and is able to process multiple data points per second. When using Apache Kafka, InfluxDB and Telegraf, MIURA is made able to also receive and report on this data in real time. However, this has not been thoroughly tested.

Requirement NF2: MIURA should be efficient enough to run on a standard consumer laptop

As specified in Chapter 8, all evaluations were performed on an old Macbook Pro. MIURA was shown to use less than 0.3GB of total available memory, and operated on a single CPU-core. This confirms that MIURA is both CPU- and memory-efficient enough to easily be deployed on consumer grade computers. When also considering the time consumption and efficiency of MIURA, running on considerably weaker hardware should also be possible, though this has not been tested.

Requirement NF3: MIURA must be able to detect anomalies in various datasets

Currently the MIURA system has only been tested on six real-world time series data. According to the evaluation results in Chapter 8, MIURA was able to identify anomalies in time series that exhibit minor fluctuations. When a time series contain high fluctuations, MIURA is unable to provide satisfactory anomaly detection. Hence, a further improvement is necessary to enhance the detection accuracy of MIURA.

Chapter 10

Discussion

This chapter will cover how this project was carried out, what experiences and decisions were made during it, and how they impacted the project positively or negatively.

10.1 Approach

10.1.1 System Development

As detailed in Appendix A, the project's goal was significantly changed from statistical forecasting and analysis to developing and implementing anomaly detection using deep learning. This drastic change greatly impacted to which capacity Orange Business was able to contribute. Additionally, both group members lacked experience in deep learning, machine learning, PyTorch, and Python. As a result, the first three weeks following the project planning had to be entirely dedicated to acquiring a foundational understanding of these fields before we could begin working on the project itself. Although this learning period delayed us by three weeks, it was crucial for gaining a better understanding of the task at hand and the project's scope.

The architecture built around MIURA, specifically the real-time monitoring, was initially planned to be addressed if there was extra time. However, it was implemented at the same time as the data streamer and database to allow proper testing of the detection models during their development.

10.1.2 Dividing Work

At the beginning of the project, during the first week of February, our supervisor Jia-Chun Lin suggested that we divide the work between the two of us. One of us would focus on implementation, handling the technical aspects of programming the detection system and implementing the additional components. The other

would focus on related work, thoroughly researching approaches similar to our own. This division of work seemed logical and allowed us to have separate responsibilities, creating a clear distinction between our tasks.

10.1.3 Thesis Writing

Finding relevant work to include in the thesis proved challenging due to our initial lack of knowledge in the field and the wide variety of approaches. We also encountered a lack of standardization in evaluation methods, including the datasets used and the correct method of calculating accuracy. This made the evaluation of the RePAD2 and MIURA approaches more challenging than anticipated.

10.1.4 Meetings and Communication

We met and communicated more or less every day. Both members live in Gjøvik which made it possible for us to meet on the NTNU campus every day of the week, with only a few exceptions. Work was discussed between us in person and decisions surrounding the project were discussed and agreed upon as they appeared. Status meetings were originally planned for Fridays, but were moved to Sundays to give us better time to complete tasks as the project became more hectic. Meeting often in person allowed us to quickly tackle and discuss situations and decisions as they appeared, leading to smoother project development. Collaborating this way also helped us gain a better understanding of what the other person was working on, giving us both a better understanding of the overarching project.

10.1.5 Use of Issue Tracker

The program used for tracking issues was changed at the beginning of the project. While Jira by Altassian seemed to be a good fit while planning the project. As we started using the program we realized that it was overly complex and slow, prompting us to switch to the built in issue board on GitHub. This worked well as it also allowed seamless integration of commits and issues.

10.1.6 Evaluation

Labeling of data

When finding datasets to evaluate the approaches on, we struggled to find good reputable labeled real-world datasets that were open-source and contained sequential anomalies. Several datasets were considered like the Yahoo Webscope datasets [97], but as these datasets require applying for access and are only open to individuals connected to academia, these were not used.

Because of the lack of good open-source labled time series datasets, through consultation with our supervisor Jia-Chun Lin we resorted to self-labeling of the UC

Irvine Machine learning Repository introduced and explained in Section 8.3. We also added a new dataset which is the cc2 dataset with changed labeling to indicate a sequential anomaly instead of a singular point anomaly. This change was done based on the labeling done by domain experts.

Evaluation process

From our understanding based on what our supervisor Jia-Chun Lin has told us in meetings, the evaluation of anomaly detection models in time series is something that is usually done manually. This is highly menial labour and we saw an opportunity to automate this process by writing and utilizing scripts to evaluate the models. Nearly every part of the evaluation process was automated including, calculating and outputting metrics, and creating graphs for visualization. Utilizing scripts for the evaluation made the process more efficient, more consistent and easier to replicate.

The scripts also simplified the process of changing how the evaluation was carried out, which proved useful as the method of evaluation needed to be changed and adjusted several times. Since the datasets used in evaluation spanned thousands of data points it could have taken several hours for each dataset, doing this for seven datasets on 6 models would have taken several days and slowed down the project considerably without taking the factor of human error into consideration. This making the inclusion scripting in the evaluation process a necessary addition to the project.

10.2 Decisions

10.2.1 Related Work

In the beginning it was planned that the related works chapter should include deep learning approaches and deep learning approaches that implement incremental learning and comparing these approaches. But halfway through the project it was decided that the section should focus on covering the broader field of anomaly detection approaches in time series. This would be done by separating the anomaly detection approaches into overarching categories of statistical, machine learning and deep learning approaches. This helps the related work cover a much broader field of anomaly detection methods and gives better understanding of the field but set the literature review back a couple of weeks. This process turned out to be more challenging than expected as papers were more often non-precise when discussing methodology and evaluation, combined with the uncertainty of how to handle methods not staying within the confines of a single method.

10.2.2 Not doing multivariate anomaly detection

The primary goal of the project was to implement a system for multivariate anomaly detection, where MIURA was meant to replace the LADs used for each stream of univariate time series in the RoLA system [98]. Further into the project we made the decision to focus on implementing the univariate anomaly detection models. This done mainly for two reasons: the fact that there would not be for proper implementation, and the fact it would not add much to the project, as the main focus was to investigate how incremental learning would impact the RePAD2 approach, making the RoLA implementation just the pure re-implementation of an existing algorithm. This allowed us to spend more time making sure RePAD2 and MIURA were properly implemented, and focus on testing those models. It should however be noted that the system developed to support the detection models was made to work with the RoLA system with only slight adjustments.

10.2.3 Dividing work

The choice of dividing work the way described in 6.2 considerably changed the way the project panned out compared to what we had originally planned. As we were a group of only two this decision did however entail some compromise, and while it meant no work overlapped, ultimately making the process more efficient, it also made it harder for us to have proper quality control of work due to the separate areas of work. This also made it so that neither of us had a fully comprehensive grasp of both the implementation and the theoretical part.

Another problem was that neither had any experience with literature review and how to carry it out in an effective and systematic manner. While many web resources shorten and explain subject matter in a short and concise manner as to appeal and be accessible to a greater audience, research papers are not written in such a way. Research papers, as was experienced, assumes that the reader already understands and possesses extensive knowledge within the respective field, making it especially hard to gather and form definitive opinions on the matter, while this chapter alone took a significant amount of time, the effort is still considered by us as necessary in order to properly put the work suggested in this project into context.

10.3 Sustainability

Today, sustainability is a key concern, as resources become more scarce it is increasingly important to develop solutions that are not only effective but also resource efficient. For this reason we believe in the importance of further developing solutions that show promise in this regard. The RePAD2 approach is already lightweight and able to run on consumer hardware, however we wanted to research further to see if the detection performance of the approach could be enhanced. The UN has set seventeen goals for sustainable development referred to as the SGDs,

and are meant to fight poverty and inequality while stopping climate change [99]. These goals serve as a global direction everyone should aim to follow, with the goal of becoming sustainable.

10.3.1 Goal 6: Clean water and sanitation

Access to clean water is essential for human well being, today it is estimated that at least one out of four people do not have access to clean drinking water [100]. An anomaly detection system such as MIURA will not be able to increase the distribution of water. However, it could be implemented to ensure the quality and safety of drinking water.

Target 6.1

Target 6.1 in the SGDs, is to provide universal and equitable access to safe and affordable drinking water to all [100]. Connected to a sensor that monitors values in water an anomaly detection system could be used to automatically detect if values are outside of what is considered normal and safe values for drinking water, notifying in real time if there is a potential problem with the water quality.

10.3.2 Goal 9: Industry, innovation and infrastructure

Goal 9 describes the importance of solid infrastructure that enables further innovation and research. Innovation and research are important as we need to further further improve old solutions or to develop new solutions to achieve sustainability [101].

Target 9.5

Target 9.5 in the SGDs discusses the importance of further scientific research and enhancing current technology [101]. The main focus of this thesis has been to further research the effects incremental learning, scalers and PyTorch have on an existing anomaly detection approach. We hope that the work we have done in this thesis will be of benefit to researchers in the field. As the work produced could give potential insight into the effect incremental learning and scalers could have on other related solutions.

10.3.3 Goal 12: Responsible consumption and production

To achieve sustainability there is a need to find solutions that enable the use of less resources to produce similar or better results, avoiding unnecessary waste. Even though an approach like MIURA will increase resource usage, it can be argued that it enables the possibility of lowering overall resource usage, by lowering waste [102].

Target 12.2

Target 12.2 describes the importance of effectively using natural resources with minimal waste. as Since RePAD2 and MIURA are such lightweight anomaly detection approaches, as discussed in the evaluation of NF2 in Chapter 9 Section 9.2. The approach will not require excessive amounts of resources to run, lowering both cost and energy usage, compared to other heavier anomaly detection systems.

Target 12.5

Target 12.5 describes the goal of reducing waste, through prevention, reduction and recycling [102]. In production machines and other hardware that experience faults or failure, anomaly detection could be implemented to gain automatic alerting of such events. This early detection of potential failure could lead to a minimization of damages reducing both cost and waste of materials. Even if the systems are not made to handle contextual anomalies they could still be effective in detecting unexpected behaviour caused by failures or faults.

10.3.4 Goal 17: Partnerships for the goals

To reach the goals set by the UN, the importance of global collaboration is emphasised. Sharing of technology and information contributes not only to the furthering of research and development it could also contributes to the building of stronger relationships and partnerships, as it allows people to build upon each other creating a strong union [103].

Target 17.6

Target 17.6 describes the betterment of collaboration on technology and innovation, to improve collective knowledge. To contribute this we decided we wanted to open-source the codebase, this includes the source code for the MIURA approach and evaluation scripts. As we see value in the work produced throughout this project, we hope that others within the field can make use of and build upon what we have done, to further the field of anomaly detection.

Chapter 11

Closing Remarks

This chapter will include overarching results of the project, learning outcome and further work will be discussed.

11.1 Learning Outcome

Before starting the project we were required to create a project plan outlining the project, how to work and what we wanted to get out of the project. We set learning goals based on what we assumed we would learn but also what we wanted to know more about. These goals turned out to be somewhat narrow compared to what we actually learned throughout the project, as we covered the goals and more.

Evaluation of deep learning models

The project evaluation was a big focus. Even though the majority of the focus was on evaluation of anomaly detection, we got a grasp on how deep learning models are evaluated.

Agile development

We have had some experience with development methods and the use of issue tracking from other courses. However, due to the larger scope of the bachelor's project, we have learned to a greater degree how crucial it is to keep track of tasks and maintain regular communication within the team.

Teamwork

Since we worked on such different parts of the project, we learned the importance of working together, communicating and transparently discussing current tasks and problems.

Tackling uncertainty

When planning the project, we were both uncertain of the steps involved in the implementation of an anomaly detection approach that uses deep learning. This uncertainty resulted in a rough generalized gantt chart. To cope with the uncertainty, we were forced to learn as we went, changing how we structured the work as necessary, making it hard to follow the Gantt chart beyond the learning phase. After the learning phase, we had to take much more hands on approach: learning as we went. This was difficult but we learned the importance of separating problems into smaller tasks and working iteratively with a focus on the task at hand, resulting in a progressively deeper understanding.

Structuring literature review

We ended up getting both familiar and comfortable with looking through literature to find the answers we were looking for. We also learned a lot when it came to structuring a literature review, and being systematic when looking through papers. As this is a process that it is easy to get lost in.

Project planning and execution

In this bachelors project, we learned the importance of proper planning. The bachelors project is the first large project us students have had to handle. In previous smaller projects planning has never been made a focus. However, in the bachelors project this is something that is required. What we experienced is that even though the project plan served an overarching outline, we did not know enough about the subjects to write a proper plan. But it was still invaluable as it forced us to think about a project outline and where to start. It also forced us to plan how to collaborate and work together throughout the project, giving us a structure to follow making it easier to stick to a routine.

Neural networks and deep learning

We ended up getting familiar with how neural networks work theoretically. We also learned how deep neural networks such as the LSTM moddel, and other simpler models are implemented and used in a practical setting, such as anomaly detection.

Python and PyTorch

Throughout the project the group had to work a lot with Python and PyTorch. This resulted in an increased proficiency in using Python for various tasks such as scripting and plotting graphs with the Matplotlib [104] library. And an increased proficiency in developing deep learning models with PyTorch.

Deep learning and real world applications

We have learned the usefulness and limitations of deep learning, as we have experienced how useful it is when processing large volumes of data and continuous monitoring. We see being able to gain a deeper insight into deep learning as a truly valuable experience, not only because of the fields current relevancy but also because of personal interest.

Streaming architecture

As we worked on the project it became more apparent what components would be necessary in the infrastructure supporting MIURA. This lead to learning about event streamers, time series databases and data handlers, and their use in an infrastructure built to support data processing.

Time series data

We were already familiar with the concept of time series data, but working with it deepened our understanding of the complex properties and dependencies this type of data contains

Anomaly detection

In the project we got familiar with anomaly detection through our own implementation and other related implementations. We also learned about the evaluation of such approaches.

11.2 Conclusion

The main goal of this project was to research if incremental learning could be used to improve upon an existing lightweight and real-time anomaly detection approach for time series data. Our solution MIURA, was found to provide higher detection accuracy compared to RePAD2, on time series data with minor fluctuations. However, it failed to accurately predict anomalies when applied to time series with higher fluctuations, highlighting areas for further improvement.

Furthermore, three different data scalers were tested during our evaluation, and was found to have significant impact on detection accuracy. According to the evaluation results, the MIURA approach was found to be highly efficient, with memory usage under 0.3GB and average detection times under 70o ms. Lastly, in an effort to contribute to the continued research within this field, all produced code, and related materials are made available to the general public as open-source at: `https://github.com/patrickjoh/miura`.

11.3 Future Work

11.3.1 Experiment with how data is standardized

The default activation functions within an LSTM model are the sigmoid and tanh functions. The sigmoid function outputs values in the range of 0 to 1, while the tanh function outputs values in the range of -1 to 1. This means that as input values become much larger or smaller than these ranges, the differences between them diminish.

For example, inputting the number 1 into a sigmoid function returns approximately 0.731, and inputting 3 returns approximately 0.952. When inputting a number like 30, it returns approximately 1, showing that large inputs lead to saturation. Saturation occurs when the output from the function becomes almost constant, causing gradients within the neural network to vanish and reducing the sensitivity of the LSTM model. This makes it difficult for the LSTM model to learn patterns in the data properly.

The current implementation of MIURA fits the standardizing scaler to the initial data used to train the LSTM model. If we assume the use of MinMaxScaler, which scales values between 0 and 1, any future data points that are significantly larger or smaller than the initial data would fall outside the optimal range of the sigmoid function. A possible solution for solving this may be to incrementally fit the scaler to accommodate new data points.

11.3.2 Make MIURA less naive

Although MIURA has demonstrated higher accuracy than RePAD2 on several time series, the current implementation has shown a tendency to train on anomalous data. When this occurs, the LSTM model that MIURA uses for predictions may become significantly less accurate. This can lead to MIURA becoming less sensitive to anomalies and more likely to train on anomalous data, further decreasing accuracy. To address this issue, a solution must be found, such as training the LSTM at regular intervals or making MIURA less likely to train on data containing anomalies. This could be achieved by enforcing a waiting period before model retraining after detecting an anomaly. Another possible solution may be to replace the LSTM model when the AARE values become too big.

11.3.3 Alerts

In a real-world situation, alerting is crucial as it enables quick and efficient handling of new anomalies. Grafana has a built-in alerting solution; however, it primarily relies on conditions meeting or exceeding certain thresholds, which does not meet our requirements. An alternative is to use Alerta [105], a flexible, self-hostable, open-source monitoring tool. Integrating MIURA with Alerta should be possible using their Python SDK or by invoking a custom Webhook, enhancing or

modifying the current reporting functionality in the code. An excessive number of alerts from an anomaly detection system can lead to the system being silenced or ignored; Alerta addresses this issue with built-in de-duplication.

11.3.4 Deployment

MIURA can be packaged into a Docker image and deployed alongside other components in the same Docker stack. It is designed to easily use environment variables, making the entire system deployable through a single Docker-compose file. Additionally, there is a need to find a way to automatically configure Grafana, including adding dashboards, data panels, and annotations for the data stream, which are currently configured manually.

MIURA can be packed into a Docker [85] image and deployed in the same stack as the other components. MIURA has already been made in a way that makes using environmental variables easy. Creating a Docker image for MIURA would make the whole system deployable through a single Docker-compose file. There is still a need to find a solution for the automatic configuration of Grafana

11.3.5 Adapt and improve documentation of code repository

The source code used in this project is open-sourced and available to the general public. However, one issue is the lack of comprehensive documentation that details the use and functionality of each component, script and program within the repository. Proper documentation is an important part of open-sourcing code, as it ensures that others can understand, use, and contribute to the project effectively. Unfortunately, there was not enough time to do this properly.

Bibliography

[1] 'Orange business.' Accessed: May 15th,2024. (), [Online]. Available: https://www.orange-business.com/en.

[2] M.-C. Lee and J.-C. Lin, *Repad2: Real-time, lightweight, and adaptive anomaly detection for open-ended time series*, 2023. arXiv: 2303.00409 [cs.LG].

[3] F. Navruzov. 'Anomaly detection for time series data: An introduction.' Accessed: January 30th, 2024. (2023), [Online]. Available: https://victoriametrics.com/blog/victoriametrics-anomaly-detection-handbook-chapter-1/.

[4] M. Munir, M. A. Chattha, A. Dengel and S. Ahmed, 'A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming data,' in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 561–566. DOI: 10.1109/ICMLA.2019.00105.

[5] A. A. Awan. 'What is incremental learning?' Accessed: May 16th,2024. (2023), [Online]. Available: https://www.datacamp.com/blog/what-is-incremental-learning.

[6] 'Pytorch.' Accessed: May 15th,2024. (), [Online]. Available: https://pytorch.org/.

[7] DeepLearning4J. 'Deeplearning4j suite overview.' Accessed: May 16th,2024. (), [Online]. Available: https://deeplearning4j.konduit.ai/.

[8] M.-C. Lee and J.-C. Lin, *Impact of deep learning libraries on online adaptive lightweight time series anomaly detection*, 2023. arXiv: 2305.00595 [cs.LG].

[9] A. Bhamidipaty, D. Patel, S. Lin, S. Jayaraman and G. Ganapavarapu. 'What is anomaly detection?' Accessed: February 1st, 2024. (2021), [Online]. Available: https://developer.ibm.com/learningpaths/get-started-anomaly-detection-api/what-is-anomaly-detection/.

[10] A. W. Services. 'What is deep learning?' Accessed: February 1st, 2024. (), [Online]. Available: https://aws.amazon.com/what-is/deep-learning/.

[11] J. Barnard and C. Stryker. 'What is anomaly detection?' Accessed: May 11th,2024. (2023), [Online]. Available: https://www.ibm.com/topics/anomaly-detection.

[12] I. El Naqa and M. J. Murphy, 'What is machine learning?' In *Machine Learning in Radiation Oncology: Theory and Applications*, I. El Naqa, R. Li and M. J. Murphy, Eds. Cham: Springer International Publishing, 2015, pp. 3–11, ISBN: 978-3-319-18305-3. DOI: 10.1007/978-3-319-18305-3_1. [Online]. Available: https://doi.org/10.1007/978-3-319-18305-3_1.

[13] C. Janiesch, P. Zschech and K. Heinrich, 'Machine learning and deep learning,' *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[14] IBM. 'What is machine learning (ml)?' Accessed: May 9th,2024. (), [Online]. Available: https://www.ibm.com/topics/machine-learning.

[15] IBM. 'What is supervised learning?' Accessed: May 9th,2024. (), [Online]. Available: https://www.ibm.com/topics/supervised-learning.

[16] IBM. 'What is unsupervised learning?' Accessed: May 9th,2024. (), [Online]. Available: https://www.ibm.com/topics/unsupervised-learning.

[17] IBM. 'What is semi-supervised learning?' Accessed: May 9th,2024. (2023), [Online]. Available: https://www.ibm.com/topics/semi-supervised-learning.

[18] IBM. 'What is reinforcement learning?' Accessed: May 9th,2024. (2023), [Online]. Available: https://www.ibm.com/topics/semi-supervised-learning.

[19] IBM. 'What is a neural network?' Accessed: May 16th,2024. (), [Online]. Available: https://www.ibm.com/topics/neural-networks.

[20] A. Krenker, J. Bešter and A. Kos, 'Introduction to the artificial neural networks,' *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, pp. 1–18, 2011.

[21] Glosser.ca, *File:colored neural network.svg*, Glosser.ca, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons, 2013. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=24913461.

[22] DeepAI. 'Weight (artificial neural network).' Accessed: May 12th,2024. (), [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network.

[23] G. for geeks. 'Backpropagation in machine learning.' Accessed: May 12th,2024. (2024), [Online]. Available: https://www.geeksforgeeks.org/backpropagation-in-machine-learning/.

[24] N. Donges. 'A complete guide to recurrent neural networks (rnns).' Accessed: May 15th,2024. (2024), [Online]. Available: https://builtin.com/data-science/recurrent-neural-networks-and-lstm.

[25] fdeloche, *File:recurrent neural network unfold.svg*, fdeloche, CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0, via Wikimedia Commons, 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg.

[26] A. Biswal. 'Power of recurrent neural networks (rnn):' Accessed: May 15th,2024. (2023), [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn.

[27] Geeks for Geeks. 'Introduction to recurrent neural network.' Accessed: May 15th,2024. (2023), [Online]. Available: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/.

[28] J. Brownlee. 'A gentle introduction to backpropagation through time.' Accessed: May 15th,2024. (2020), [Online]. Available: https://machinelearningmastery.com/gentle-introduction-backpropagation-time/.

[29] S. Patil. 'Vanishing gradient problem in rnns.' Accessed: May 15th,2024. (2023), [Online]. Available: https://plainenglish.io/community/vanishing-gradient-problem-in-rnns-9d8e14.

[30] DeepAI. 'Exploding gradient problem.' Accessed: May 15th,2024. (), [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem.

[31] S. Hochreiter and J. Schmidhuber, 'Long short-term memory,' *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[32] fdeloche, *File:long short-term memory.svg*, fdeloche, CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0, via Wikimedia Commons, 2017. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=60149410.

[33] C. Olah. 'Understanding lstm networks.' Accessed: May 15th,2024. (2015), [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[34] Geeks for Geeks. 'Understanding of lstm networks.' Accessed: May 15th,2024. (2023), [Online]. Available: https://www.geeksforgeeks.org/understanding-of-lstm-networks/.

[35] Scylla. 'Time series data.' Accessed: May 11th,2024. (), [Online]. Available: https://www.scylladb.com/glossary/time-series-data/.

[36] Dataiku. 'Understanding time series data.' Accessed: May 15th,2024. (2024), [Online]. Available: https://doc.dataiku.com/dss/latest/time-series/understanding-time-series.html.

[37] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018, p. 31, https://otexts.com/fpp2/tspatterns.html#tspatterns(Accessed: May 12th,2024).

[38] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018, pp. 221–230, https://otexts.com/fpp2/stationarity.html(Accessed: May 12th,2024).

[39] A. Tate. 'Understanding the importance of stationarity in time series.' Accessed: May 12th,2024. (2023), [Online]. Available: https://hex.tech/blog/stationarity-in-time-series/.

[40] V. Chandola, A. Banerjee and V. Kumar, 'Anomaly detection: A survey,' *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009, ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. [Online]. Available: https://doi.org/10.1145/1541880.1541882.

[41] F. Palmieri and U. Fiore, 'Network anomaly detection through nonlinear analysis,' *Computers & Security*, vol. 29, no. 7, pp. 737–755, 2010, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2010.05.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404810000362.

[42] K. Doshi, S. Abudalou and Y. Yilmaz, 'Reward once, penalize once: Rectifying time series anomaly detection,' in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9891913.

[43] 'Apache kafka.' Accessed: May 16th,2024. (), [Online]. Available: https://kafka.apache.org/.

[44] AWS. 'What is apache kafka?' Accessed: May 17th,2024. (), [Online]. Available: https://aws.amazon.com/what-is/apache-kafka/.

[45] 'Telegraf.' Accessed: May 15th,2024. (), [Online]. Available: https://www.influxdata.com/time-series-platform/telegraf/.

[46] 'Influxdb.' Accessed: May 15th,2024. (), [Online]. Available: https://www.influxdata.com/.

[47] Stackhero. 'Influxdb: Introduction.' Accessed: May 17th,2024. (), [Online]. Available: https://www.stackhero.io/en/services/InfluxDB/documentations/Introduction.

[48] 'Grafana.' Accessed: May 15th,2024. (), [Online]. Available: https://grafana.com/.

[49] J. Walker. 'What is grafana and when should you use it?' Accessed: May 17th,2024. (2022), [Online]. Available: https://www.howtogeek.com/devops/what-is-grafana-and-when-should-you-use-it/.

[50] 'Meta open source.' Accessed: May 16th,2024. (), [Online]. Available: https://opensource.fb.com/.

[51] NVIDIA. 'Pytorch.' Accessed: May 16th,2024. (), [Online]. Available: https://www.nvidia.com/en-us/glossary/pytorch/.

[52] J. Sawtell-Rickson. 'What is pytorch?' Accessed: May 17th,2024. (2022), [Online]. Available: https://builtin.com/machine-learning/pytorch.

[53] A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, 'Machine learning for anomaly detection: A systematic review,' *IEEE Access*, vol. 9, pp. 78 658–78 700, 2021. DOI: 10.1109/ACCESS.2021.3083060.

[54] A. H. Yaacob, I. K. Tan, S. F. Chien and H. K. Tan, 'Arima based network anomaly detection,' in *2010 Second International Conference on Communication Software and Networks*, 2010, pp. 205–209. DOI: 10.1109/ICCSN.2010.55.

[55] W. Liu, H. Jiang, D. Che, L. Chen and Q. Jiang, 'A real-time temperature anomaly detection method for iot data.,' in *IoTBDS*, 2020, pp. 112–118.

[56] LinkdIn. 'Luminol.' repo for the anomaly detection library Luminol. (2023), [Online]. Available: https://github.com/linkedin/luminol/commits/master/.

[57] J. Hochenbaum, O. S. Vallis and A. Kejariwal, *Automatic anomaly detection in the cloud via statistical learning*, 2017. arXiv: 1704.07706 [cs.LG].

[58] A. Siffer, P.-A. Fouque, A. Termier and C. Largouet, 'Anomaly detection in streams with extreme value theory,' in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 1067–1075.

[59] I. Bose and R. K. Mahapatra, 'Business data mining — a machine learning perspective,' *Information & Management*, vol. 39, no. 3, pp. 211–225, 2001, ISSN: 0378-7206. DOI: https://doi.org/10.1016/S0378-7206(01)00091-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037872060100091X.

[60] S. Zhong, S. Fu, L. Lin, X. Fu, Z. Cui and R. Wang, 'A novel unsupervised anomaly detection for gas turbine using isolation forest,' in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2019, pp. 1–6. DOI: 10.1109/ICPHM.2019.8819409.

[61] F. T. Liu, K. M. Ting and Z.-H. Zhou, 'Isolation forest,' in *2008 eighth ieee international conference on data mining*, IEEE, 2008, pp. 413–422.

[62] N. Laptev, S. Amizadeh and I. Flint, 'Generic and scalable framework for automated time-series anomaly detection,' in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15, Sydney, NSW, Australia: Association for Computing Machinery, 2015, pp. 1939–1947, ISBN: 9781450336642. DOI: 10.1145/2783258.2788611. [Online]. Available: https://doi.org/10.1145/2783258.2788611.

[63] S. Oehmcke, O. Zielinski and O. Kramer, 'Event detection in marine time series data,' in *KI 2015: Advances in Artificial Intelligence: 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings 38*, Springer, 2015, pp. 279–286.

[64] 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*, 2000, pp. 93–104.

[65] R. K. Malaiya, D. Kwon, S. C. Suh, H. Kim, I. Kim and J. Kim, 'An empirical evaluation of deep learning for network anomaly detection,' *IEEE Access*, vol. 7, pp. 140 806–140 817, 2019. DOI: 10.1109/ACCESS.2019.2943249.

[66] J. Audibert, P. Michiardi, F. Guyard, S. Marti and M. A. Zuluaga, 'Usad: Unsupervised anomaly detection on multivariate time series,' in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20, Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 3395–3404, ISBN: 9781450379984. DOI: 10.1145/3394486.3403392. [Online]. Available: https://doi.org/10.1145/3394486.3403392.

[67] M. Munir, S. A. Siddiqui, A. Dengel and S. Ahmed, 'Deepant: A deep learning approach for unsupervised anomaly detection in time series,' *IEEE Access*, vol. 7, pp. 1991–2005, 2019. DOI: 10.1109/ACCESS.2018.2886457.

[68] E. Li, Y. Li, S. Bedi, W. Melek and P. Gray, 'Incremental learning of lstm-autoencoder anomaly detection in three-axis cnc machines,' *The International Journal of Advanced Manufacturing Technology*, vol. 130, no. 3, pp. 1265–1277, 2024.

[69] T. Wen and R. Keyes, *Time series anomaly detection using convolutional neural networks and transfer learning*, 2019. arXiv: 1905.13628 [cs.LG].

[70] M.-C. Lee, J.-C. Lin and E. G. Gran, 'Repad: Real-time proactive anomaly detection for time series,' in *Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, Springer, 2020, pp. 1291–1302.

[71] N. Zou, J. Wang, G.-L. Chang and J. Paracha, 'Application of advanced traffic information systems: Field test of a travel-time prediction system with widely spaced detectors,' *Transportation Research Record*, vol. 2129, no. 1, pp. 62–72, 2009. DOI: 10.3141/2129-08. eprint: https://doi.org/10.3141/2129-08. [Online]. Available: https://doi.org/10.3141/2129-08.

[72] influxdata. 'Sql data types.' Accessed: May 16th,2024. (), [Online]. Available: https://docs.influxdata.com/influxdb/cloud-serverless/reference/sql/data-types/.

[73] L. DALY. 'Kanplan: Where your backlog meets kanban.' Accessed: May 16th,2024. (), [Online]. Available: https://www.atlassian.com/agile/kanban/kanplan.

[74] 'Microsoft teams.' Accessed: May 16th,2024. (), [Online]. Available: https://www.microsoft.com/nb-no/microsoft-teams/log-in.

[75] 'Signal.' Accessed: May 16th,2024. (), [Online]. Available: https://signal.org/nb/.

[76] 'Discord.' Accessed: May 16th,2024. (), [Online]. Available: https://discord.com/.

[77] 'Overleaf.' Accessed: May 18th,2024. (), [Online]. Available: https://www.overleaf.com/.

[78] 'Onedrive.' Accessed: May 16th,2024. (), [Online]. Available: https://www.microsoft.com/nb-no/microsoft-365/onedrive/online-cloud-storage?market=no.

[79] 'Github.' Accessed: May 18th,2024. (), [Online]. Available: https://github.com/.

[80] Altassian. 'Jira.' Accessed: May 18th,2024. (), [Online]. Available: https://www.atlassian.com/software/jira.

[81] 'Clockify.' Accessed: May 18th,2024. (), [Online]. Available: https://clockify.me/.

[82] 'Conventional commits 1.0.0, A specification for adding human and machine readable meaning to commit messages.' Accessed: January 29th, 2024. (), [Online]. Available: https://www.conventionalcommits.org/en/v1.0.0/#summary.

[83] A. Singh, *Kafka-tig*, https://github.com/eternalamit5/Kafka-TIG/commits/main/, 2024.

[84] 'Docker compose overview.' Accessed: May 18th,2024. (), [Online]. Available: https://docs.docker.com/compose/.

[85] 'Docker overview.' Accessed: May 20th,2024. (), [Online]. Available: https://docs.docker.com/get-started/overview/.

[86] 'Plugin directory.' Accessed: May 20th,2024. (), [Online]. Available: https://docs.influxdata.com/telegraf/v1/plugins/.

[87] J. Brownlee. 'How to scale data for long short-term memory networks in python.' Accessed: May 18th, 2024. (2019), [Online]. Available: https://machinelearningmastery.com/how-to-scale-data-for-long-short-term-memory-networks-in-python/.

[88] Scikit-learn. 'Compare the effect of different scalers on dtaa with outliers.' Accessed: May 14th,2024. (), [Online]. Available: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html.

[89] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, 'Scikit-learn: Machine learning in Python,' *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[90] influxdata. 'Python client library.' Accessed: May 14th,2024. (), [Online]. Available: https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/python/.

[91] scikit-learn. 'Sklearn.preprocessing.robustscaler.' Accessed: May 20th,2024. (), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html.

[92] scikit-learn. 'Sklearn.preprocessing.standardscaler.' Accessed: May 20th,2024. (), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[93] S. Ahmad, A. Lavin, S. Purdy and Z. Agha, 'Unsupervised real-time anomaly detection for streaming data,' *Neurocomputing*, vol. 262, Jun. 2017. DOI: 10.1016/j.neucom.2017.04.070.

[94] S. Vito, *Air Quality*, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C59K5F, 2016.

[95] M. E. A. Sehili and Z. Zhang, *Multivariate time series anomaly detection: Fancy algorithms and flawed evaluation methodology*, 2023. arXiv: 2308.13068 [cs.LG].

[96] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong and Q. Zhang, 'Time-series anomaly detection service at microsoft,' in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, ACM, Jul. 2019. DOI: 10.1145/3292500.3330680. [Online]. Available: http://dx.doi.org/10.1145/3292500.3330680.

[97] yahoo. 'Webscope datasets.' Accessed: May 18th,2024. (), [Online]. Available: https://webscope.sandbox.yahoo.com/.

[98] M.-C. Lee and J.-C. Lin, *Rola: A real-time online lightweight anomaly detection system for multivariate time series*, 2023. arXiv: 2305.16509 [cs.LG].

[99] FN-SAMBANDET. 'Fns bærekraftsmål.' Accessed: May 19th, 2024. (2024), [Online]. Available: https://fn.no/om-fn/fns-baerekraftsmaal.

[100] FN-SAMBANDET. 'Rent vann og gode sanitærforhold.' Accessed: May 19th, 2024. (2023), [Online]. Available: https://fn.no/om-fn/fns-baerekraftsmaal/rent-vann-og-gode-sanitaerforhold.

[101] FN-SAMBANDET. 'Industri, innovasjon og infrastruktur.' Accessed: May 19th, 2024. (2023), [Online]. Available: https://fn.no/om-fn/fns-baerekraftsmaal/industri-innovasjon-og-infrastruktur.

[102] FN-SAMBANDET. 'Ansvarlig forbruk og produksjon.' Accessed: May 19th, 2024. (2023), [Online]. Available: https://fn.no/om-fn/fns-baerekraftsmaal/ansvarlig-forbruk-og-produksjon.

[103] FN-SAMBANDET. 'Samarbeid for å nå målene.' Accessed: May 19th, 2024. (2023), [Online]. Available: https://fn.no/om-fn/fns-baerekraftsmaal/samarbeid-for-aa-naa-maalene.

[104] 'Matplotlib.' Accessed: May 18th,2024. (), [Online]. Available: https://matplotlib.org/.

[105] 'Alerta.' Accessed: May 18th,2024. (), [Online]. Available: https://alerta.io/.

Appendix A

Project Plan

NTNU

Kunnskap for en bedre verden

DCSG2900 - Bachelor's thesis in Digital Infrastructure og Cybersecurity

Real-Time Multivariate Anomaly Detection using Deep Learning

Project Plan

Authors:
Magnus Westerheim Johannessen, Patrick Bjørkhaug Johannessen

01.02.2024

Table of Contents

1 BACKGROUND AND GOALS

1.1 Background

Orange Business, formerly Basefarm, is a leading provider of mission-critical IT solutions. Knowing that Modern IT infrastructures rely heavily on the real-time monitoring and analysis of system metrics to ensure optimal performance, security, and reliability we were first tasked to develop a solution for forecasting and predicting future trends based on historical data using Telegraf, InfluxDB and Graphana. However, after reviewing the assignment with our supervisors we concluded that either additions or changes to the original assignment had to be made. More clearly differentiate ourselves from other theses, and because the tasks would be too easily accomplished. One of our supervisors, Jia-Chun Lin, suggested a new direction for the project: implementing a real-time anomaly detection system using deep learning, based on her state-of-the-art research, particularly the papers on RePAD2 [1] and RoLA [2].

Given Orange Business's role in providing mission-critical IT solutions, the implementation of a robust anomaly detection system is vital. Such a system is essential for maintaining the functionality and integrity of critical IT infrastructure, meaning that a high-performance, real-time and lightweight anomaly detection system would not only benefit Orange Business but also a wide range of businesses and devices reliant on data. The role of an anomaly detection system is to serve as an early warning system, alerting to potential issues such as unusually high resource usage, sudden temperature increases, or malfunctioning sensors. Therefore, accurate and precise anomaly detection is key to timely addressing potential issues. However, the effectiveness of such systems relies on high-accuracy as "a model that creates too many false alarms is not helpful because it constantly disrupts regular operation routines, prompting users to "silence" such tools." [3]. Traditional systems primarily use statistical classification and regression models, and while statistical models excel for time series forecasting, they often struggle with adapting to trends, seasonality and change-points [4]. This being the reason to instead use deep learning to implement our anomaly detection system.

1.2 Project Goals

The project's goals have been divided into three categories: result goals, impact goals and learning goals. Result goals are what the group aims to achieve, in the form of concrete results and deliverables. Impact goals are the reason behind the project and what the group wants the stakeholder to get out of the project. Learning goals are the learning outcomes the group hopes to gain during the duration of the project.

Result Goals

The solution will accurately detect anomalous data in real-time while having a low margin for false positives, allowing for greater awareness of anomalies in time series data.

1. Provide insight into potential improvements to how the individual anomaly detection models work and are trained.
2. A model that is able to accurately predict anomalies in a multivariate time series with an F-score and time consumption that is either comparable or better than the original model.

Impact Goals

The final model should be able to aid companies in accurately detecting suspicious activity in time series data using minimal resources and without the problem of resource exhaustion. Additionally, it should grant greater insight in the use of neural networking in anomaly detection.

1. Increased awareness around potential anomalies and suspicious activity.
2. Better ensure the stability of critical systems and applications through monitoring.
3. Provides companies with accurate and real-time anomaly detection.

Learning Goals

Gaining a deeper understanding of topics such as machine learning, infrastructure and agile development methodologies

1. Learn what neural networks are, how they work, and how they are implemented.
2. Increased understanding of machine learning and the real world applications.
3. Learn and understand how different types of neural networks differ and their applications.
4. Gain understanding of how to work with machine learning and deep learning.
5. Gain insight and understanding of how machine learning models are developed and evaluated.
6. Better understand how agile development methods work and how they can be applied in a real-world context.

2 SCOPE

2.1 Subject Area

- Anomaly detection
  - Anomaly detection refers to methods for identifying points in data that deviates from expected behaviour [5].
- Multivariate time series
  - A multivariate time series refers to observations of two or more variables taken from a device or a system simultaneously over time, that may or may not be correlated [2].
- Deep learning
  - Deep learning is a subset within Artificial Intelligence (AI) that attempts to simulate the behaviour of the human brain using neural networks to process large amounts of data [6].

2.2 Task Description

The objective of the assignment is to research whether a system like the RoLA [2] would benefit from the usage of incremental learning. As of now the RoLA system retrains itself from scratch when failing to accurately predict the next data point.

To determine this two systems will be developed, one like RoLA that makes use of retraining and one that uses incremental learning. Both the Lightweight Anomaly Detectors (LAD's) [1] and the RoLA system were originally written in the programming language Java. Since the group does not have any prior experience with Java, our implementation will be written in Python. To provide a reliable comparison, the two models will be evaluated using the same publicly available multivariate time-series datasets.

2.3 Limitations

- Types of anomaly detection - To reduce complexity the anomaly detection system will be made to strictly detect point anomalies and collective anomalies, not contextual anomalies.
- Evaluation metrics - Evaluation of all models will be limited to the following metrics, in order to provide a reliable comparison:
  - Precision
  - Recall
  - F-Score
  - Average training time
  - Average detection time
  - Average total time consumption
- Publicly available datasets - Evaluation of both models will be limited to only existing publicly available data-sets.
- Framework - PyTorch will be the as the primary framework used for implementing the deep learning models, and other frameworks like TensorFlow will not be considered. This decision is made because PyTorch was shown to provide the best overall efficiency and accuracy with RePAD, of which RePAD is directly based on [7].
- RoLA and *Lightweight Anomaly Detection* - We limit ourselves to base the anomaly detection system on the already existing RoLA system and the LAD's (Lightweigth Anomaly Detection) models it makes use of.

2.4 Possible Additions

If there is time or opportunity, additions may be made to supplement the quality, representation or scope of the project. Possible additions will be thought out and decided on further into the project, with more knowledge and a better contextual foundation to back up these decisions.

As of now, examples of possible additions are:

- Using InfluxDB and Grafana to visualize results outputted from the models in real time.
- Implementing the model in a simulated enterprise environment.

3 PROJECT ORGANIZATION

3.1 Roles and Responsibilities

Project work is equally divided between the group's members, using a co-management structure.

- **Stakeholder** - Truls Enstad
  - Represents Orange Business during the project
- **Co-leader** - Patrick Bjørkhaug Johannessen
  - Primary responsibility for ensuring accurate and proper recording of time and tasks in the designated systems.
- **Co-leader** - Magnus Westerheim Johannessen
  - Primary responsibility for ensuring all deadlines are maintained, both internal and external.
- **Shared responsibilities** - Both Co-Leaders
  - Shared overarching responsibility
  - Quality-control of all completed tasks and related documentation.
  - Preparations for meetings and the subsequent writing of meeting reports.
  - Maintaining an overview of the project and steering work in the right and agreed upon direction.
  - Upholding the set weekly schedule.
  - Communication with stakeholder and supervisors.
  - Clear and reasonable storing of all documents, files or other material .

3.2 Rules and Routines

3.2.1 Routines

- Report: The report, amongst other documents, are written in Overleaf.
- Backup of report: The report's source files will be backed up to OneDrive weekly.
- Source code: All source code shall reside in dedicated Github repositories.
- File storage: OneDrive acts as our primary storage platform and is to be used for all documents and files that does not naturally reside in either Overleaf or Github.
- Internal communication: Signal is the primary platform for all internal communications.
- Communication with supervisors: Microsoft Teams is the established platform for all communications with our supervisors.
- Time tracking: All hours are tracked into designated projects/categories using Clockify, at least daily.
- Issue tracking: Jira is used for both management and tracking of all tasks/issues, which is expected to be updated continuously throughout the day.
- Weekly review: Every week a meeting is scheduled, of which the primary purpose is to review what was done the previous week.
- Digital meetings: When physical meetings are not possible, Discord will be used for voice and video calls.
- Continuous quality control: All work performed by either group member must be reviewed by the other member before considered done.

3.2.2 Rules

Work Procedures

- Every member has to average a minimum of 30 hours spent on the project per week, which approximates to 540 hours spent on the project and is the agreed upon minimum. Official vacation weeks are exempt from this.
- The group expects to physically meet 5 days a week to work on the project.
- All members of the group are expected to log hours worked and work done, and in the agreed upon manner and systems.
- All group members are required to attend all scheduled meetings. This includes internal status meetings, meetings with supervisors and meetings with the stakeholder.
- Meeting minutes must be written for all meetings.

Delivery

- Members of the group share the responsibility for work being delivered within due dates.
- Deliverables are always to be properly reviewed by both members whereas the one responsible for delivery should review them before submitting, while the other member reviews the submitted deliverables.

Absence

- If possible, notice is to be given at least 24 hours in advance in cases where a member of the group is unable to meet to the agreed upon time or at all.
- Should any internal conflicts arise within the group, this is to first be attempted resolved internally. If this fails to be handled internally, the thesis supervisors will be notified.
- Should unforeseen circumstances cause possible long-term absence of any group member, the member in question is obligated to notify the group of this at the earliest possible time.
- Should a team member be required to spend considerable time at home or elsewhere due to long term sickness or other circumstances, the group will adapt to the extent that it is possible.
- In the event that a member of the group is no longer able to work on the thesis, the member in question is expected to inform the group's supervisors. In cases where this is not possible, the same is expected of the other member.

4 PLANNING, FOLLOW-UP AND REPORTING

4.1 Development Model

Machine learning and deep learning are fields where neither of us has any past experience, and the steps involved in implementing the anomaly detection system are therefore highly uncertain and unknown. This makes it difficult to outline a clear path or set fixed goals, as we anticipate adjustments and learning curves throughout the project's development. This called for a methodology and framework that was flexible enough to handle continuous change and lightweight enough to not hinder progress. The choice fell on Kanplan, a mixed methodology that combines Kanban's visual approach, continuous work cycle and focus on tasks, with the backlog from Scrum. This approach allows for a near pure Kanban workflow while avoiding an overwhelming and extensive "To-Do" list, ensuring that prioritized tasks remain at the forefront.

While Kanban functions with a high degree of autonomy, the backlog in Kanplan needs active prioritization. Rather than a designated role, the group conducts a weekly review to assess and update the priorities of each task in the backlog. The priorities used are primarily "High", "Medium", "Low", whereas "Highest" and "Lowest" may be used on special occasions. Any task with priority above "Medium" is automatically placed in the "To-Do" column of the Kanban-board. If all tasks with a priority of "High" or above are completed, tasks with a priority of "Medium" are to be picked individually from the backlog until the next weekly review. During each of these reviews there will be made a status report concerning the group's progress, problems and priorities that is sent to our supervisors.

The Kanban board is split into the columns "To-Do", "On-Hold", "In-Progress", "Review" and "Done", whereas "On-Hold" is meant for prioritized tasks that are waiting for something else, while "Review" is for all tasks that the assignee has completed and is waiting to undergo review by the other group member. All task-related work is tracked using Jira, where tasks are added and consistently updated with information regarding their progress.

Workflow

- A backlog of all currently known tasks
- Weekly review where the priorities of all tasks are set or updated, deciding the primary focus of the coming week.
- All tasks on the board are then assigned, giving the group members temporary ownership of those tasks.
- A completed task is first placed in "Review", ready to undergo quality control by the other group member before finally being placed in "Done".
- Throughout the week:
  - If one member of the group finishes their assigned tasks while the other member still has unfinished tasks, a discussion can be held about taking over the task.
  - If all tasks on the board are either completed or in progress, new tasks will be picked individually from the backlog starting with those of "Medium" priority.
- At the next review, unfinished tasks stay on the board while new ones are added from the backlog and assigned.

4.2 Scheduled Meetings

- Meetings with our supervisors are scheduled every Tuesday from 14:30 to 15:00.
- Meetings with the stakeholder are held bi-weekly and are scheduled on Thursdays every even-numbered week, from 13:00 to 13:30.
- Internal group status meetings are held every Thursday from 15:00 to 16:00. This meeting's main function is to review what work has been completed, assess and update the priorities of each task in the backlog, and to write the weekly status report that is submitted to the supervisors.

5 ORGANIZATION OF QUALITY ASSURANCE

5.1 Documentation, Standards and Source Code

5.1.1 Configuration Management

- All source code created during the project is stored in their respective Github repositories.
- Reports and meeting minutes are written in LaTeXwith the web-based collaboration platform Overleaf.
- All other documents related to the project are stored in a shared folder on OneDrive.
- The source code of the project report is backed up to OneDrive every week during the weekly review.
- All files contained in OneDrive will be backed up weekly to an external SSD during the weekly review.

All listed services have implementations of version control, providing features like rolling back to an earlier version of a file. This is helpful in cases like files becoming corrupt or "damaged" as a result of human error. OneDrive is used as the main storage for all files related to the project due to how clearly Microsoft communicates the importance of high availability and data resiliency, and how they ensure this for their services. [8]

5.1.2 Documentation

Documentation serves as a fundamental part of our project and is the most important thing we do, even more so than the actual work. Without it, development becomes complicated and the resulting thesis, unreliable. Writing of proper documentation is therefore an important part of the project and is expected for all completed work. Documentation includes, but is not limited to, clear and concise commenting and documentation of code, choices made during development, methods and results.

To maintain these requirements, all related documentation is reviewed alongside the actual work to address the cognitive blindness that arises from familiarity and extensive knowledge of the completed tasks.

5.2 Development and Work Routines

- All work is defined and added to the backlog in Jira.
- A maximum of 1 or 2 related tasks are allowed to be worked on simultaneously, per member.
- *Commit* messages follow the standard specification of Conventional Commits. [9]
- The description of all commits must contain the *issue key* of the relevant issue in Jira.
- All written code should be commented in such a manner as to clearly convey it's purpose and how it functions.
- All code is commented before being *commited*.
- Naming of functions, variables and similar shall adhere to the standard guidelines for the relevant language and framework.
- All completed tasks goes through a *Review* by the other member before it is marked as resolved.

Workflow in Jira

Figure 1: Workflow in Jira

Issue-status

Status	Description
To do	Tasks not yet started
On Hold	Task is on hold, and is waiting for another task.
In Progress	Tasks being worked on
Review	Tasks is finished and waiting for review
Done	Task is done

Issue-labels

Label	Description
administrative	Related to administrative functions or structure
change	Changes or additions
development	Suited for changes or remarks related to development.
documentation	Improvements or additions to documentation
functionality	New feature or change of functionality
question	Further information is requested
quickfix	Issues that can be done in under 10 minutes
time_consuming	Extra time and attention is needed.
unconfirmed_issue	Observed problem or possible bug requiring further investigation
reopened	Issue previously resolved, now needing further attention.

Tools

5.3 Evaluation of Models

In order to properly compare and measure our anomaly detection models, some standard metrics must be used. These metrics are:

- F-score, calculated by the amount of false positives and the amount of false negatives
- Time consumption, including the times it takes to train new models and the time it takes for a model to detect anomalies.

The Numenta Anomaly Benchmark (NAB) contains multiple labeled datasets for evaluating real-time anomaly detection. Therefore, these datasets were initially considered to be used for this, but were shown to not provide multivariate metrics [10]. Other datasets featuring multivariate metrics must therefore be sought out.

5.4 Risk Analysis

The group identified the biggest risks surrounding the thesis by rating their level of probability and severity. To decide probability and severity the group members played risk poker as a starting point, then discussed the results to ascertain levels of severity and probability.

Probability	**Severity**				
	Minor	**Marginal**	**Moderate**	**Severe**	**Critical**
Certain	Medium	High	High	Very high	Very high
Likely	Medium	Medium	High	High	Very high
Possible	Low	Medium	Medium	High	High
Unlikely	Low	Medium	Medium	Medium	High
Rare	Low	Low	Low	Medium	Medium

5.4.1 Risks

Conflict within the group	
Description	Internal disputes that go beyond normal discussion.
Probability	Unlikely
Severity	Marginal
Risk	Medium
Measures	Taking a break, either prolonged or a short food break

Deadlines not upheld	
Description	Work not delivered within the given deadline.
Probability	Rare
Severity	Critical
Risk	Medium
Measures	Delivery dates are marked in calendars with notifications. Work is done with the internal schedule in mind to ensure steady progress towards deadlines.

Loss of data	
Description	Data loss would imply the loss of work or otherwise relevant documents/information surrounding the project work.
Probability	Unlikely
Severity	Critical
Risk	High
Measures	The group uses tools with implemented version control like Overleaf and Github, in addition to backing up all project work to both OneDrive and physical storage

Scope creep	
Description	Continuously increasing the scope of the original assignment to the point that the final product becomes incoherent and incomplete.
Probability	Unlikely
Severity	Severe
Risk	Medium
Measures	Clarify the requirements of the project, clearly define the scope of the assignment and set additional features with defined boundaries that can be implemented in the event that the main product is completed.

Loosing a team member(permanently)	
Description	The event where a team members is forced to or chooses to leave the group and discontinue the project work
Probability	Rare
Severity	Critical
Risk	Medium
Measures	To prevent a team member leaving the group communication will be vital, should an issue arise it is required that all team members are made aware of the issue so that a solution can be reached. If relevant, supervisors will be contacted. To reduce the consequences of a group member leaving both members have access to all work that is done. The supervisors will be contacted and the remaining group member will continue the project.

Sickness or other temporary absences	
Description	Team member becomes ill and is unable to attend project work at campus. Leave duration: 1-5 work days.
Probability	Likely
Severity	Minor
Risk	Medium
Measures	To reduce the consequences of illness within the group, the sick team member will contact the other member at the first available time and inform them of the situation. The team member that is ill will also work from home to the extent that it is possible.

Long term absence	
Description	Long term absence could involve serious illness that forces a team member to stay away from campus for an extended period of time. 4 weeks or longer is considered an extended period of time.
Probability	Rare
Severity	Critical
Risk	Medium
Measures	The group will go over to an online approach so that the project progress is minimally affected.

Product is not fully realised	
Description	The problem described in the assignment description is not fully realised/solved to completion.
Probability	Possible
Severity	Minor
Risk	Low
Measures	To prevent an incomplete product/solution the group will define clear milestones to ensure progress, the group will also use a Kanban board to have a better overview of tasks and what state those tasks are in. To reduce the consequences of the product not being in a finished working state the group will document all work that is done during the project.

Internal schedule not upheld	
Description)	The teams internal milestones/deadlines are not met.
Probability	Unlikely
Severity	Severe
Risk	Medium
Measures	Proper project management and a set workflow is key. Larger tasks will be broken down into smaller more manageable tasks. The team members hold each other accountable. Take regular breaks to keep productivity high.

6 TIMELINE

6.1 Gantt

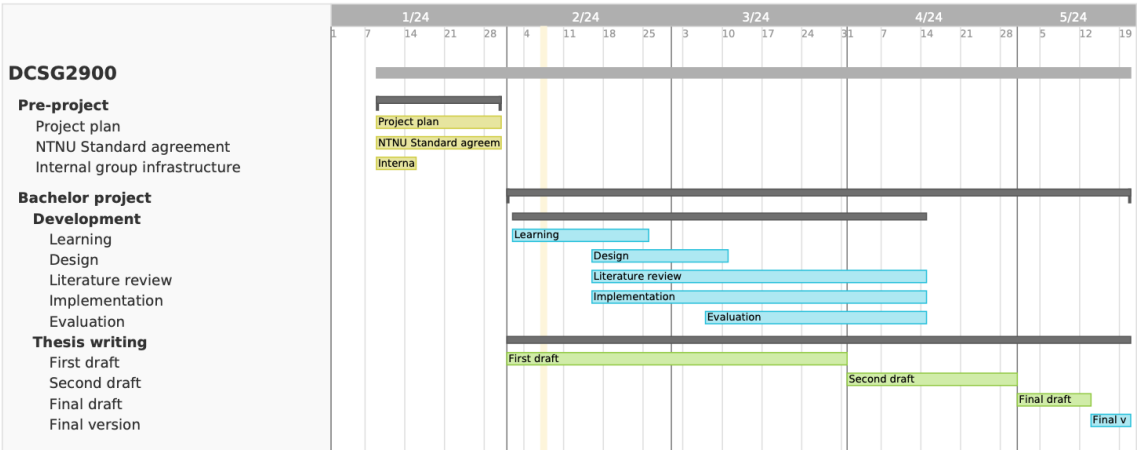


Figure 2: Gantt

- **Pre-project**
  - **January 9th to February 1st:** Project plan
  - **January 9th to February 1st:** NTNU Standard agreement
  - **January 9th to January 15th:** Internal group infrastructure
- **Bachelor project**
  - **Development**
    - **February 2nd to February 25th:** Learning
    - **February 16th to March 10th:** Design
    - **February 16th to April 14th:** Literature review
    - **February 16th to April 14th:** Implementation
    - **March 7th to March 31thth:** Evaluation
  - **Thesis writing**
    - **February 1st to March 31th:** First draft
    - **April 1st to April 30th:** Second draft
    - **May 1st to May 13th:** Final draft
    - **May 14st to May 20th:** Final version

References

[1] M.-C. Lee and J.-C. Lin, *Repad2: Real-time, lightweight, and adaptive anomaly detection for open-ended time series*, 2023. arXiv: 2303.00409 [cs.LG].

[2] M.-C. Lee and J.-C. Lin, *Rola: A real-time online lightweight anomaly detection system for multivariate time series*, 2023. arXiv: 2305.16509 [cs.LG].

[3] F. Navruzov. 'Anomaly detection for time series data: An introduction'. Accessed: January 30th, 2024. (2023), [Online]. Available: https://victoriametrics.com/blog/victoriametrics-anomaly-detection-handbook-chapter-1/.

[4] M. Munir, M. A. Chattha, A. Dengel and S. Ahmed, 'A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming data', in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 561–566. DOI: 10.1109/ICMLA.2019.00105.

[5] A. Bhamidipaty, D. Patel, S. Lin, S. Jayaraman and G. Ganapavarapu. 'What is anomaly detection?' Accessed: February 1st, 2024. (2021), [Online]. Available: https://developer.ibm.com/learningpaths/get-started-anomaly-detection-api/what-is-anomaly-detection/.

[6] A. W. Services. 'What is deep learning?' Accessed: February 1st, 2024. (), [Online]. Available: https://aws.amazon.com/what-is/deep-learning/.

[7] M.-C. Lee and J.-C. Lin, *Impact of deep learning libraries on online adaptive lightweight time series anomaly detection*, 2023. arXiv: 2305.00595 [cs.LG].

[8] Microsoft Corporation. 'Sharepoint and onedrive data resiliency in microsoft 365'. Accessed: January 29th, 2024. (2023), [Online]. Available: https://learn.microsoft.com/en-us/compliance/assurance/assurance-sharepoint-onedrive-data-resiliency.

[9] 'Conventional commits 1.0.0, A specification for adding human and machine readable meaning to commit messages'. Accessed: January 29th, 2024. (), [Online]. Available: https://www.conventionalcommits.org/en/v1.0.0/#summary.

[10] S. Ahmad, A. Lavin, S. Purdy and Z. Agha, 'Unsupervised real-time anomaly detection for streaming data', *Neurocomputing*, vol. 262, Jun. 2017. DOI: 10.1016/j.neucom.2017.04.070.

Glossary

Clockify Online tool for time tracking. https://clockify.me. 5

Discord A social platform for messaging and voice- and video-calls. https://discord.com. 5

Github Online service for hosting of Git-repositories. https://github.com. 5, 9, 12

Grafana Grafana is a platform for creating and sharing dashboards that combine data from multiple sources, such as logs, metrics, and traces. https://grafana.com. 4

InfluxDB InfluxDB Open Source is a single binary that delivers time series data collection, processing, and analysis. https://www.influxdata.com/products/influxdb/. 4

Jira Online software development tool, used for it's digital Kanban board and issue tracking. https://www.atlassian.com/software/jira. 5, 7, 9, 10

Kanban Visual approach to project management. https://www.atlassian.com/agile/kanban. 7

Kanplan A combination of Kanban and a backlog (as used in Scrum). https://www.atlassian.com/agile/kanban/kanplan. 7

Microsoft Teams Online service for collaboration, communication and meetings. https://www.microsoft.com/en-us/microsoft-teams/group-chat-software. 5

NAB An anomaly detection benchmark and testing framework for real-time streaming data. https://www.numenta.com/resources/htm/numenta-anomoly-benchmark/. 11

OneDrive Online platform for storing, sharing and collaborating on files and documents. https://www.microsoft.com/en-gb/microsoft-365/onedrive/online-cloud-storage. 5, 9, 12

Overleaf Online collaboration platform for writing and compiling of Latex-documents. https://overleaf.com. 5, 12

PyTorch Open Source framework for machine learning and deep learning https://pytorch.org/. 3

Scrum A very focused framework for project management based around sprints of set length. https://www.atlassian.com/agile/scrum. 7

Signal An encrypted and open-source messaging application for instant messaging. https://signal.org. 5

TensorFlow Open Source framework for machine learning and deep learning https://www.tensorflow.org. 3

Name	Date	Signature
Patrick B. Johannessen	01.02.2024	Patrick Johannessen
Magnus W. Johannessen	01.02.2024	Magnus Johannessen

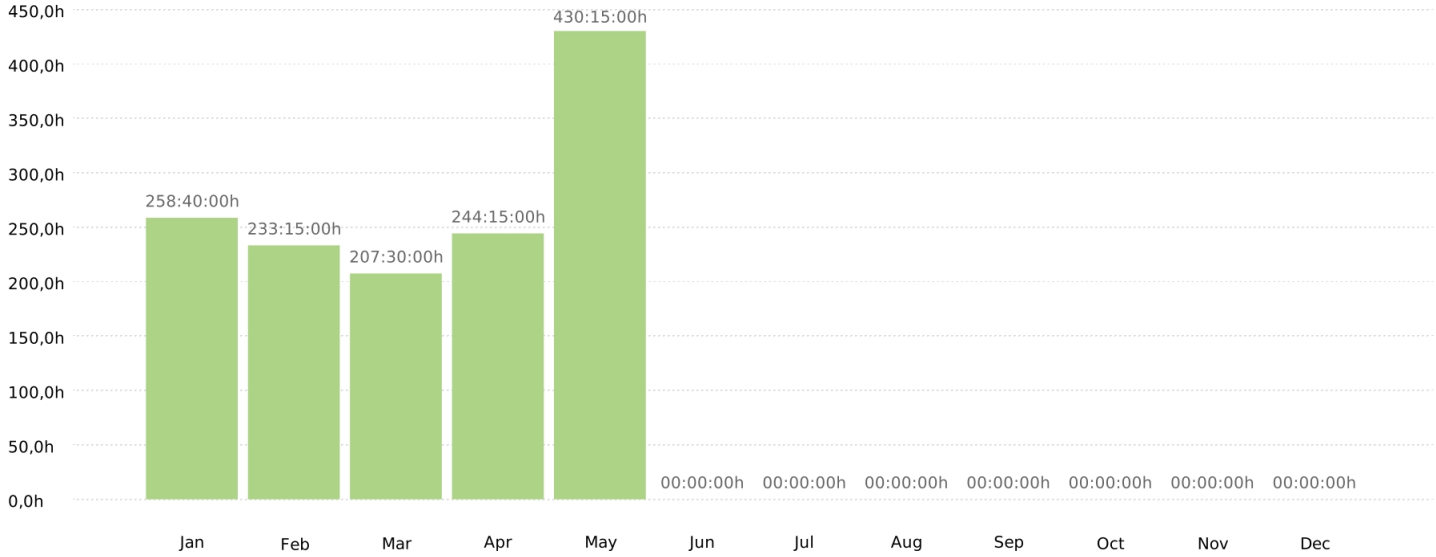
Appendix B

Timekeeping

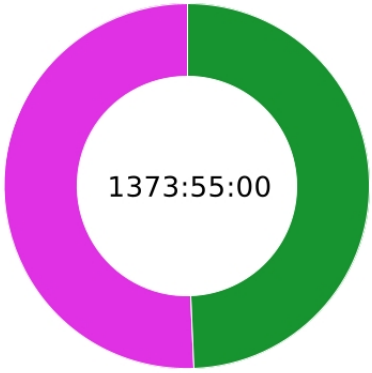
Summary report

01/01/2024 - 31/12/2024

Total: 1373:55:00 Billable: 00:00:00 Amount: 0,00 USD

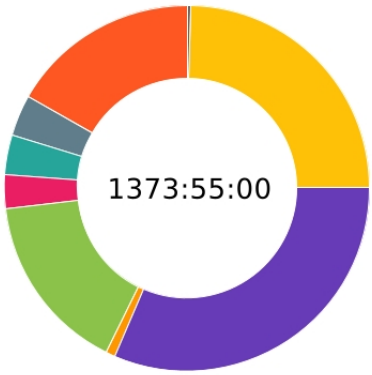


User



Magnus Johannessen	695:10:00	50,60%
Patrick Johannessen	678:45:00	49,40%

Project



Coding	229:30:00	16,70%
Documentation	49:50:00	3,63%
Evaluation	47:45:00	3,48%
Meetings	41:10:00	3,00%
Project Plan	219:10:00	15,95%
Reflection	10:45:00	0,78%
Report	433:15:00	31,53%

Research/Learning	338:30:00	24,64%
Seminar	04:00:00	0,29%

User / Project	Duration	Amount
Magnus Johannessen	695:10:00	0,00 USD
Coding	21:15:00	0,00 USD
Documentation	23:50:00	0,00 USD
Meetings	20:55:00	0,00 USD
Project Plan	116:25:00	0,00 USD
Reflection	10:15:00	0,00 USD
Report	247:45:00	0,00 USD
Research/Learning	253:45:00	0,00 USD
Seminar	01:00:00	0,00 USD
Patrick Johannessen	678:45:00	0,00 USD
Coding	208:15:00	0,00 USD
Documentation	26:00:00	0,00 USD
Evaluation	47:45:00	0,00 USD
Meetings	20:15:00	0,00 USD
Project Plan	102:45:00	0,00 USD
Reflection	00:30:00	0,00 USD
Report	185:30:00	0,00 USD
Research/Learning	84:45:00	0,00 USD
Seminar	03:00:00	0,00 USD

Appendix C

Meeting Reports

10.01.2024 - Start-up meeting with Orange Business

Participants

- From Orange Business:
  - Truls Enstad
- Team members:
  - Patrick Bjørkhaug Johannessen
  - Magnus Westerheim Johannessen
- Other students:
  - Kristina Sætre Katakis
  - Eskil Lykke Refsgaard
  - Elias Rudsbråten

Goals

Increase understanding for the given assignment and the assignment description

Agenda

1. Clarify the assignment description
2. Plan future meetings.

Notes and further work

- Hva slags tjeneste er det vi skal sette opp forecasting for? - Han er ikke helt sikker, men det som går an er det at vi definerer et par spørsmål også sende de på mail, men forecastingen skal funke generelt. Vi bestemmer mye selv - Hvilken type trafikk? - Hvilke komponenter skal arkitekturen ha/hvordan skal den se ut? - Ingen vedlagt konfigurasjon i ny oppgavetekst

- Hvilke scenario skal det testes for? «1. Simulate various system loads and scenarios to validate the forecasting models.»

- Antar at dette er feil? «2. Introduce intentional system disruptions to evaluate the efficacy of the anomaly detection framework.” - Simulering av feil

- Hva ble vi enige om angående tjenester som skal kjøres/testes (Web-server, mail-server, kubernetes-cluster og hvorfor?, MySQL-Database)

- I hvilken grad skal vi ta høyde for incident response measures? Altså at det skal dokumenteres mulig respons dersom det forutsies høy trafikk? «2. Create comprehensive reports highlighting forecasting results, and incident response measures. »

- Både langtidsanalyse/forecasting og korttidsanalyse/forecasting? Altså, skal det predikeres hvorvidt ressursbruk vil overgå tilgjengelige ressurser, og gi varsel om

dette, både i korte tidsrom (Som konsekvent økning over et tidsrom på 1-2 timer), og i lange tidsrom (Samme bare over uker og måneder?)

- Er meningen å ende opp med noe som er ment å bli sett på rent visuelt, eller som er «produkt» som også selv «gir lyd fra seg» om visse parametere/scenarioer blir møtt?

- Løsningen vil hjelpe predikere når komponenter som f.eks disker fylles opp - Både korttids og langtids, funke også som et varslingssystem - Lagring av hendelser/loggføres disord varsel

16.01.2024 - Meeting with supervisor

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Få klarhet rundt oppgaven og oppgaveteksten.

Agenda

1. General information concerning the supervisors roles in the thesis.
2. Chosen language for the thesis
    - If we decide to write the thesis in English can we still present the thesis in Norwegian?
3. Is it possible for us to make changes to the project plan post delivery date(31st of January)
4. Are we supposed to create a collaboration agreement and then reference it in the project-plan?
5. For future meetings, is it best to have them with both supervisors present? or is it better to have them with one supervisor at a time.
6. We are unsure how to write/explain roles and responsibilities as we are a small group(2 members)
7. Suggestions for internal meeting structure.

Notes and further work

- Go through and cross off checklist from supervisors
- Finish "NTNU standardavtale"
- The project plan must be reviewed by supervisors prior to submission. Preferably twice before the final deadline.
- Translate parts of the thesis and related documents to English
- Deadline for first draft of the thesis is 31st of March
- Endelig presentasjon kan gjøres på norsk, men må være forberedt på engelske spørsmål og svar
- Thesis can be presented in Norwegian, but we need to be prepared for questions in English and to answer in English.
- We have to write weekly status reports, the reports will be handed in at least one day before meeting with the supervisors.
- Investigate the need for designated roles, and possibly designated type of work

- Improve rules and procedures concerning absence and worst case scenarios
- Get acquainted with ntnu.no/ub and Google Scholar for eventual research articles.
- Consider/explore possible extensions to the original assignment (e.g. 1. automatic scaling of resources based on predictions made. 2. visualization of response time for users when scaling).
- Elaborate progress plan

23.01.2024 - Meeting with supervisor(s)

Participants

- Jia-Chun Lin (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Update supervisors on our progress, get feedback on draft and .

Agenda

1. Are we specifically graded on the project plan or is it more as a help to us? not graded
2. Standard agreement delivery, another group delivered in our channel
3. Sign standard NTNU agreement
4. Are we supposed to create a collaboration agreement and then reference it in the project-plan?
    - Do we need one or should we just put all of the rules etc in the project plan
5. Is the project plan finalized after 31st of January or are we allowed to make changes?
6. Discuss content and structure of "2. Scope"
7. Look at the overall structure of the report
8. Discuss structure and "essential" content in "5. Organisation of quality assurance". Unsure about "Testing" and "Routines".
9. Anomaly detection group wants to share infrastructure etc.
   evaluation, open source dataset to evaluate our approach, evaluating the approach using different metrics, F-score, time consumption needs to be evaluated(how quickly anomalies can be detected and determined) how to fit orange into the picture of the proposed anomaly detection

Notes and further work

- Kelly suggested that the addition to the task should not be automatic scaling (because it has been done many times before and is documented in several previous reports. It would also be a lot to set up if one were to try to differentiate further). She therefore suggested expanding the task with anomaly detection using convoluted neural networks and an unsupervised model for continuous training, which is thus online and dynamically learning. The model therefore improves with use and does not require a new version with updated training data to detect previously unseen "anomalies".

- Supervisor(Kelly) proposed anomaly detection based on machine learning(convoluted neural networks) as an addition to the assignment.
- The anomaly detection is not trained in an offline environment rather the machine learning model is trained in an online environment with "real data".
- Evaluation part of the project plan should include(if we take the anomaly detection further and Orange approves) using open source datasets to evaluate the approach, evaluating by using different metrics, F-score, time consumption needs to be evaluated(how quickly anomalies can be detected and determined).
- We need to figure out what Telegraf can do, what is its limitations. What data it can collect etc.
- We need to propose the anomaly detection extension of the assignment and get it approved by Orange Business.
- The task description part of the project plan should include not what Orange states in the assignment description but rather what we plan to do.
- Research if neural networks(convoluted etc) and Long Short Term Memory(LSTM)
- Homework: Read papers sent over by Kelly and decide whether or not we want to do the anomaly detection
- Finish the project plan well within deadline, 28th of January.

25.01.2024 - Meeting with Orange Business

Participants

- From Orange Business:
  - Truls Enstad
- Team members:
  - Patrick Bjørkhaug Johannessen
  - Magnus Westerheim Johannessen

Goal

Figure out what should be done with the assignment, has to be changed?

Agenda

1. What does Orange think of changing parts of the assignment, doing anomaly detection instead of forecasting?
2. Høre hva de tenker om at vi går fra forecasting til anomaly detection. Skiller oss fra den andre gruppen ved at vår deteksjon blir drevet av maskinlæring fremfor satte regelsett
3. What does Orange think about us changing the task from forecasting to anomaly detection. Considering that we differentiate ourselves from the other group by that our detection runs on machine learning instead of pre-defined rules.
4. What should be kept from our original task description? We can for example still build the infrastructure for data generation and can still use Telegraf to feed Apache kafka.
5. Will this be ok? or do you have any other suggestions in terms of an extension to the original assignment? in case it is not ok.

Notes and further work

- We should listen to and prioritize feedback from our supervisor
- In terms of the assignment, the group's learning and final report is the priority.
- As a group we should not focus too much on if Orange benefits from our assignment or not, as a good project/report is the most important part.
- Orange wants to know the details of our new approach. The group needs to send a document describing why the changes were necessary and what those changes entail. This should also include why a new approach is necessary and what the primary changes will be.
- Could our group also tackle anomaly detection? This would mean both groups have the same primary objective (As both groups have the same

stakeholder). Would this be a problem considering the vastly different approaches? (Our supervisor gives a clear no.)

30.01.2024 - Meeting with supervisor(s)

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Få klarhet rundt oppgaven og oppgaveteksten.

Agenda

1. Discuss whether project plan is final or changeable throughout the project
2. Sign the NTNU standard agreement.
3. Orange wanted us to write and send over documentation/specification of what the changes are and what they involve.
4. Further discuss the "new" approach/new assignment.
5. Feedback on the project plan.
6. How to fit Orange into the new task description
7. Could Orange potentially intervene and deny the change of our task?
8. Would Orange gain access to the system at the end of the bachelors project?
9. Ask about sections we were unsure about (Background, Limitations, Timeline/Gantt, Additional features)

Notes and further work

- General: The project plan can be progressively changed throughout the project, and all details need not be set in stone. This is mostly regarding to details that are as of now hard to clearly define.
- General: Telegraf should not be necessary to include as we do not plan to simulate and generate our own data to feed into the model. Instead open source data-sets will be used for evaluation of the models. Implementing the deep learning models is our primary focus, and it's also beneficial to not do this in order to more clearly separate us from the other group.
- General: We need to discuss internally what we want to do with the source code, should it be open source and should we hand it over to Orange?
- General: The report itself should and will be public, but it may be beneficial to keep the source code "closed". It will still reside in a Github-repository, but as private.
- Task description: We should focus on implementing/developing the deep learning system. Creating and implementing an environment to simulate network traffic is unnecessary. The models with be evaluated using open-source data-sets. May be a potential addition.

- Task description: We base base our project on trying to making a model with and one without incremental learning in order to compare which works best.
- Limitations: The limitations-section should not discuss the actual implementation or details regarding .
- Limitations: To limitations, remove the now unnecessary elements (Ubuntu and OpenStack) and add that we limit ourselves to only focus on point anomalies and collective anomalies, but not contextual anomalies (Like environment factors happening outside the actual data).
- Limitations: A possible limitation is that we limit ourselves to only use the Pytorch or Tensorflow framwork (whereas we will probably decide on PyTorch).
- Additional features: This can be added to later as until we have worked with it and learned more it is hard to see what would be good additions.
- Additional features: Probably smart to include as standard part of the project is a real-time visualization of trends and detections using Influxdb as the database and dashboards like Chronograf and Graphana.
- Gantt: We should flesh out the Gantt with additions like "Learning period", "Design period", "Implementation period". Testing should be renamed to "Evaluation". In general, the Gantt-diagram does not need to be overly detailed and some parts can and will naturally overlap.
- Gantt: Evaluation and verification will be taking a lot of time (More than two weeks) and so the Gantt diagram should be adjusted to account for this. 3 weeks?

06.02.2024 - Meeting with supervisor(s)

Participants

- Jia-Chun Lin (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Clarify the way forward and improvements to the project plan

Agenda

- Recommended educational resources on deep learning.

Summary

-

Notes and further work

- Look through and review datasets further:
    - Check whether Numenta Anomaly Benchmark (NAB) contains multivariate and labeled data sets
    - Look thorough the labeled dataset with monitored human activity provided by Kelly.
    - Prolong the planned timelines for both implementation and evaluation of the models.
    - Add literature survey to the Gantt, parallel to implementation. Not necessarily 1 to 1 + update planned time according to the above mentions point.
- Fix minor typos and implement other points of improvements shown by Kelly (Jia-Chun Lin):
    - Provide explanation on why choices are made i.e PyTorch instead of TensorFlow.
    - Citing of sources should be done in before the period/ending of the actual sentence. something something [1].
    - Refer to sources like research papers for choices made and why they were made. For example referring to a paper on why PyTorch is better for our use than TensorFlow.
- Other:
    - Look and learn from PyTorch example code, instead of creating and learning everything from scratch. (I.e implementation and training of LSTM-models)

- It may be a good idea to distribute personal workloads, Kelly suggested splitting the work into "literature review" and "implementation".
- Chapter 1 of the thesis will be based on the project plan.
- Do a closer review of the previewed example report
- Nice to implement the visualization using InfluxDB and Grafana, if time. Nice addition and to stay a bit closer to the original assignment.

09.02.2024 - Meeting with Orange Business

Participants

- From Orange Business:
  - Truls Enstad
  - Mohammad Reza Jafari
- Team members:
  - Patrick Bjørkhaug Johannessen
  - Magnus Westerheim Johannessen

Goal

Make sure Orange has a clear understanding of the new assignment.

Agenda

1. Clarify the new/updated assignment.
2. Explain the specifics of the new assignment.
3. Talk about our thoughts of the way forward and what we wish to realize.
4. Questions from Truls.

Notes and further work

- Went through the new assignment details and how it deviates from the initial assignment.
  - Anomaly Detection, but using deep learning instead of a rule based approach as with ADTK.
  - Is not resource intensive (Lightweight Deep Learning model).
  - Does not require dedicated training or training data.
  - Continuously learns unsupervised and online, in real-time.
  - Dedicated infrastructure for collecting "realistic" data is not longer part of the project.
  - Data used for evaluating the model will be publicly available labeled datasets made for evaluating real-time anomaly detection model systems.
  - Hope to include the visualization of the model's trends and detection results using InfluxDB and Grafana.
- If we require reasoning/approval on why we changed the assignment Orange is more than happy to provide. It won't be a problem.
- When it comes to questions surrounding the new assignment it will be difficult for Orange to provide assistance, Questions like this will be directed towards our supervisor.

- Truls og Orange vil gjerne ha en skriftlig beskrivelse/oversikt over hva oppgaver innebærer og hvilke endringer som er gjort. Ble enige om sende prosjektplanen.

20.02.2024 - Meeting with supervisors

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Get feedback and input on our progress so far

Agenda

1. Tips on how to structure the design period?
2. Recommended reading materials/papers
3. Recommended learning material/ subjects we should look at
4. Go through RePAD in more detail

Notes and further work

- Supervisor(Kelly) advised us to not spend as much time on the fundamentals but to rather start implementing the models we are to work on. This because we risk having too little time to spend on evaluation and thesis writing.
- Kelly went over how the current RePAD2 model retrains itself and how incremental learning will work
- When possible, models should not be coded from scratch, but rather use existing code found on thr web.
- Kelly will herself look over and have her student take a look at some material potentially useful for the implementation stage.
- Kelly would like us to implement an LSTM-model by this week
- Each LAD(RePAD2) would have their own thread, aka. work concurrently. (Should all work in parallel). Must find out how concurrency works in Python.

28.02.2024 - Meeting with supervisors

Participants

- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Gain better clarity on the task and task description.

Agenda

- Present progress on implementing the LSTM model and RePAD2.
- Discuss the current implementation
- Discuss our relative progress

Notes and further work

- Showcased the LSTM-model and explained the code. Elaborated further on parts Ameen was unsure of, for example scaling of the data with the min/max scaling
- Ameen meant an LSTM-model should be simple
- Using a different learning rate from the model used by Kelly and her partners shouldn't be a problem as learning rate depends on data.
- Talked about which optimizers have been used/tested (Adam and SGD)
- We explained to Ameen that the model uses 3 previous datapoint to predict 1 future data point and moved on to explain where the LSTM model fits into the larger picture in terms of RoLA and RePAD2
    - Went over presentation on RoLA
    - Discussed how our implementation will Differ to the one suggested in the RoLA paper by Kelly and her partner.
- As the LSTM model seems to work best with a dataset with a linear trend it was suggested to linearize the data so that the model works, but it seemed unlikley to be relevant
- Alot of the project is similar to the paper that Kelly published so Ameen seemed unsure of what is different, this may be indication that if time presents itself we should add to the project.

05.03.2024 - Meeting with supervisors

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Agenda

- Present progress on implementing the LSTM model and RePAD2.
- Discuss the current implementation
- Discuss our relative progress

Notes and further work

- A datastreamer will input datapoints into a database, Kelly and her colleague used mySQL, they did this as it would not be practical or functional to save datapoints in memory.
- Displayed and explained current progress on our LSTM-model
- Explained current progress on the RePAD2-algorithm and discussed how data should be handled and processed
- Datastreamer (Kafka) should be connected to the database (We are thinking InfluxDb) and not the program itself
- Data is set up to be normalized, but this may not work properly because it could make current data incomparable against later data.
-
- The database should keep a record of all data-points, AARE until the limit (Queue)
- Detected anomalous points in time should be recorded in order to visualize it
- Kelly mentioned that RePAD2 is the most important part of the project, but also the most challenging. When that part is done the rest should be relatively easy except for incremental learning.
- Can use Numenta Anomaly Benchmark for testing RePAD2 and the Ferry dataset to test RoLA. The same datasets as was used in the research papers should be used in order to properly compare our implementations with theirs.
- For later visualizations (If time) it would be very beneficial to be able to show detected anomalies in real-time through a dashboard like Grafana.
- Kelly will send a paper on why they concluded with 3 datapoints for prediction, She mentioned that it was because using more datapoints did not improve the prediction but increased the processing requirements.

- Something to look at towards the end (Again, if time) is reducing the amount of false positives that RePAD2 and RoLA generates at the beginning. We may experience the same, and this may be a good thing to look at.
- Kelly will give us a paper/explanation on why and how the tanh activation function is implemented in RePAD2.
- RePAD2 is not suitable for seasonal patters as it does not retain enough memory to do so, it will also be interesting to see if incremental learning will improve RePAD2's ability to predict seasonal anomaly.

07.03. - Title

Participants

- From Orange Business:
  - Truls Enstad
  - Sebastian Strømnes
- Team members:
  - Patrick Bjørkhaug Johannessen
  - Magnus Westerheim Johannessen

Goal

Get orange up to speed on what has been done

Agenda

1. Explain to orange our current progress

Notes and further work

- The lstm model is finished and work on the model that decides if a datapoint is an anomaly or not(RePAD2) is coming along nicely
- Mentions to Truls that we will be using both Apache Kafka and InfluxDB, and how this will be implemented into the RePAD2 model. Truls mentions that Orange already uses Apache Kafka in their system so its a nice addition to the stack.
- Its good that the implementation is multivariate as this is relevant to their systems.
- Its good that we use labeled datasets to simulate real world data as it means the implementation can easily be implemented into for example their systems
- We explain our challenges with the task, such as learning deep learning and python.
- The task is roughly explained to Sebastian as he joins the meeting.
- Patrick explains further how incremental learning is the main focus of the task.
- Truls agrees that learning before implementing is the best way to go about it. Learning is also important when it comes to the thesis.

19.03.2024 - Meeting with supervisors

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Get input from Kelly on the existing RePAD2 implementation

Agenda

- Showing off the current progress on RePAD2
- Show off our system for automatically annotating anomalies

Notes and further work

- Explained and demonstrated the current progress on the RePAD2-model to Kelly and Ameen
    - Visualization of both data and anomalies in real-time
    - How events from datasets sent from Kafka and automatically sent to and stored in InfluxDB
    - explains how data travels from csv -> Kafka -> Telegraf -> InfluxDB -> MIURA
    - How the model queries and handles event from InfluxDB
    - The remaining part of the RePAD2 code is the RePAD2-specific logic, Kelly agrees that this should not be a problem.
    - How alerts are easy to implement with either InfluxDB or Grafana
- Kelly sees our work so far as sufficient and suggested that we should focus on implementing both RePAD2-models instead of using time on RoLA as well. This lends us more time for evaluation and potential improvements to both models.
- In order to fairly evaluate the models there should be used a set margin on either side of anomalous timestamps. As long as the model detects an anomaly anywhere within that range of events it is considered a true positive.
- The "Related work"-part of the thesis will include the literature review, the review of existing models and arguments why or why not the model we have suits us best. To mention some.
- "Methodology" part of the thesis should explain the architecture, why the stack was put together that way and explanation of each service/part of the stack.

- "Evaluation" should include the same metrics as detailed in the research paper for RePAD2. This part should also include the average time it takes to retrain the model, the average time it takes to detect an anomaly, how many times the model had to be retrained, the average time it takes to decide that a data point is not anomalous and the average time it takes to decide on any given event.
- We should use more than just the two datasets used in the reaearch paper for RePAD2. At least four different datasets should be evaluated.
- Should test the model with the different scaler-types (MinMaxScaler, RobustScaler, StandardScaler) and without normalizing the data to see how and whether this impact performance

21.03.2024 - Meeting with Orange Business

Participants

- From Orange Business:
  - Truls Enstad
  - Terje Nomeland
  - Mohammad Reza Jafari
- Team members:
  - Patrick Bjørkhaug Johannessen
  - Magnus Westerheim Johannessen

Goal

Discuss the new task with the person that wrote the original task

Agenda

1. go through our assignment for the person who wrote the original task description
2. go through the visual representation of the anomalies

Notes and further work

- The person that wrote the original task(Terje) notes that the most important thing is that we manage to do and utilize predictions and baselines, in terms of the original task
- Continue to explain that the implementation to Terje:
  - We use the nerual network model lstm for predicting 1 datapoint in the future based on the 3 previous datapoints
    - For the lstm model the framework pytorch is used, Terje agreed that this was a good framework as it is highly documented, fast and has low latency(queries are almost instant)
  - The implementation is based on our supervisors papers, mostly RePAD2
  - The RePAD2 model that makes use of the lstm models prediction does calculations such as threshold and AARE score to determine if the predicted datapoint is an anomaly
  - Terje mentioned that its good to keep computations simple an fast as it allows weaker hardware to run the model
  - Noted that the pytorch part of the code is complete and what remains on the RePAD2 implementation are calculations
  - The model when completed will be lightweight, online and unsupervised

  - Terje notes that if it is desirable to predict datapoint further into the future
  - Terje then asks if we cleanl the data, we do not but calculate AARE scores to see if datapoints are within the threshold. Patrick also goes over when recalculations and retraining is done and why.
  - Go over with Terje that the main focus of the project will be to implement incremental into the model.
  - Retraining the model is beneficial as it allows the model to adapt to newer data
  - They had a system that was trained offline and would not adapt to changes in data a "static" model. But wasn't practical
  - To display the visualization an image of the visual output with anomalies annotated was sent to orange.

- Tips from Orange:
  - Make the baseline solution good rather than splitting the focus and you could reflect on future work in the bachelor thesis.
  - When presenting it might be helpfull to have a demo ready to show to the sensors.
  - When presenting, if you do a live demo, prerecord it so save yourself the trouble of technical issues in the demonstration. Using a usb-drive to keep such videos on is also a good idea.
  - State alarms for anomalies would be nice, especially if an anomaly had passed and the state was clear. This would be a benefit as it would make troubleshooting easier.
  - also having a system that ranks the anomalies by severity would also be cool to see.

26.03.2024 - Internal meeting

Participants

- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Discuss the report structure

Agenda

- Go over the thesis structure
- should the chapter of development process be there and should it be between method and implementation
- Is Orange still the project owner
- what is expected of the first draft(expected level of completeness)
- Should the discussion part be separated into(results from the models, comparing them and discussing the project and decisions we took)

Notes and further work

- Went over the thesis structure.
- The discussion part of the report should focus on reflection, there should be a separate part that discusses the results.
- Talked through parts of RePAD2-algorithm that Patrick thought to be unclear.
- Were unsure about why T == 7 was unique (Line 12 of the pseudo code), but understand now that D_T+1 is predicted until T == 7, whereas predicting a new value is skipped when T == 7. This makes it so that from now on D_T is predicted. T == 7 did not need to be predicted within the else-if at line 11 as the prediction for T == 7 was made at line 10.

02.04.2024 - Meeting with supervisors

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Both discuss and talk about the finished models and the first draft of the report

Agenda

1. Talk about the structure and get feedback from supervisors
2. Talk about the logic behind the incremental-learning version of RePAD2
3. Discuss possible ways to make the models better
4. Show and discuss visualizations of detections made by the two models from datasets CC2 and B3B

Notes and further work

- Thesis:
    - 1.1 Should be called project background
    - Task description should probably be after project background
    - Subsubsections within scope should just be their own subsections
    - Neural networks in "Background" should probably be a subsection under "Deep learning"
    - The "Background" section should have more of a linear progression in terms of the subsections, for example time series -> anomaly detection -> deep learning
    - Related work should have more overarching categories, such as statistical, machine learning, deep learning.
    - Don't necessarily need separate section for Forecasting or model training
    - Keep in mind not to make the related work section too long to avoid tiring the sensor, but also not too short.
    - For the use of research papers "Four or five papers per category would be fine", as said by Kelly.
    - "Method" should be mostly about the architecture and the choices behind how we built it and why we built it that way.
    - In "Method", pseudocode for RePAD2 and the incremental version should be included like in the research paper for RePAD2.

    - For the "Implementation", include how things was used, how components are connected, what kind of services and what versions was used (What Python verson was used etc)

- Anomaly detection models:
    - The lack of detection by the model in the beginning of the dataset could be because of an inconsistency in the code as the original RePAD2 model is quite sensitive.
    - The range that is predicted as an anomaly becomes smaller and smaller, could this mean that the model learns on the anomalous data?
    - The incremental model as it is now could be the most "naive" version of the implementation and could be improved with for example check if it still trains on anomalous data.
    - Thesis should mention that both the models work in real time, and so does the visualization and annotation of anomalies.
    - Should test the model without normalization, and other scalars
    - Will send Kelly the code so she could look at it
    - For presentation a pre-recording of the real-time anomaly detection.

04.04.2024 - Meeting with Orange Business

Participants

- From Orange Business:
  - Truls Enstad
- Team members:
  - Patrick Bjørkhaug Johannessen
  - Magnus Westerheim Johannessen

Goal

Update Truls on progress

Agenda

1. Update Orange on our progress since last time

Notes and further work

- Explains to Truls that since last time both models have been made, and are currently being further worked on to fix some issues in the code. Then further explains that the models don't detect ranges but rather singular points.
- Talked about how the model learns and that it might also learn anomalous data
- Then further discussed possible how to fix this and further extensions that might solve the problem
- Could be helpful in some cases to use granular data to detect other anomalies
-

05.05.2024 - Meeting with Kelly and Leo

Participants

- Jia-Chun Lin/Kelly (Supervisor)
- Ming-Chang Lee/Leo (Research colleague of Kelly)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Discuss with Kelly and Leo to gain better understanding of the code/pseudocode of RePAD2

Agenda

1. Discuss the model and its printed AARE values and threshold then compare it to Leo's models printed values

Notes and further work

- The results of the model looks nice but Kelly and Leo would like to know why the model is less sensitive, are the results because of pytorch or is it a problem in the logic of the code.
- Leo suggest to test the model on a recurrent time series dataset to see if the model detects an anomaly only the first time it occurs and not later times. If it only detects the anomaly the first time there is a problem.
- Patrick show the raw output of the model(printed values of AARE and prediction) and Kelly comments on the threshold is high in the beginning
- Leo displays his output of AARE and threshold visualized and eventually raw format from the mysql database
- Leo further explain the process of the model, at what datapoint AARE values are calculated and at what point threshold and anomaly detection starts. From this it is apparent that the way the model calculates is not the same way Leo's model does it.
- Where the model Starts, if it is at 0 or 1 does not matter as long as the logic makes sense and is the same.
- When adapting the model to be as Leo and Kelly's save the previous one, instead of changing it.

09.04.2024 - Meeting with supervisors

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

show off

Agenda for next meeting

- Point adjust
- (Related work) How to categorize models that uses a combined approach i.e deep learning and statistical.
- Show off
- Have a look at their results (With AARE-values)

Notes and further work

- The baseline model that is similar to RePAD2 does not have any false positives, the reason for this could be the fact that the LSTM-model written in PyTorch is better at predicting earlier on in the time series. This also produce a better threshold early on. AARE are then more likely to be kept beneath the threshold if not an anomaly.
- (Evaluation) We should not use the duplicated time series data when evaluating in the thesis
- (Evaluation) There should be done further comparison of the incremental and baseline models by running them on more datasets. Could also use some artificial Numenta-datasets.
- (Evaluation) Bigger/Longer datasets datasets than. Optimally more than 8000 data points.
- (Evaluation) As one of the models only use one AARE-value to determine the threshold it would also be interesting to see a model that experiments with more than 3, what happens when increasing the amount of AARE-values used?
-

16.04.2024 - Meeting with supervisors

Participants

- Jia-Chun Lin (Supervisor)
- Ameen Chilwan (Supervisor)
- Magnus Westerheim Johannessen
- Patrick Bjørkhaug Johannessen

Goal

Gain clarity surrounding related work and datasets

Agenda

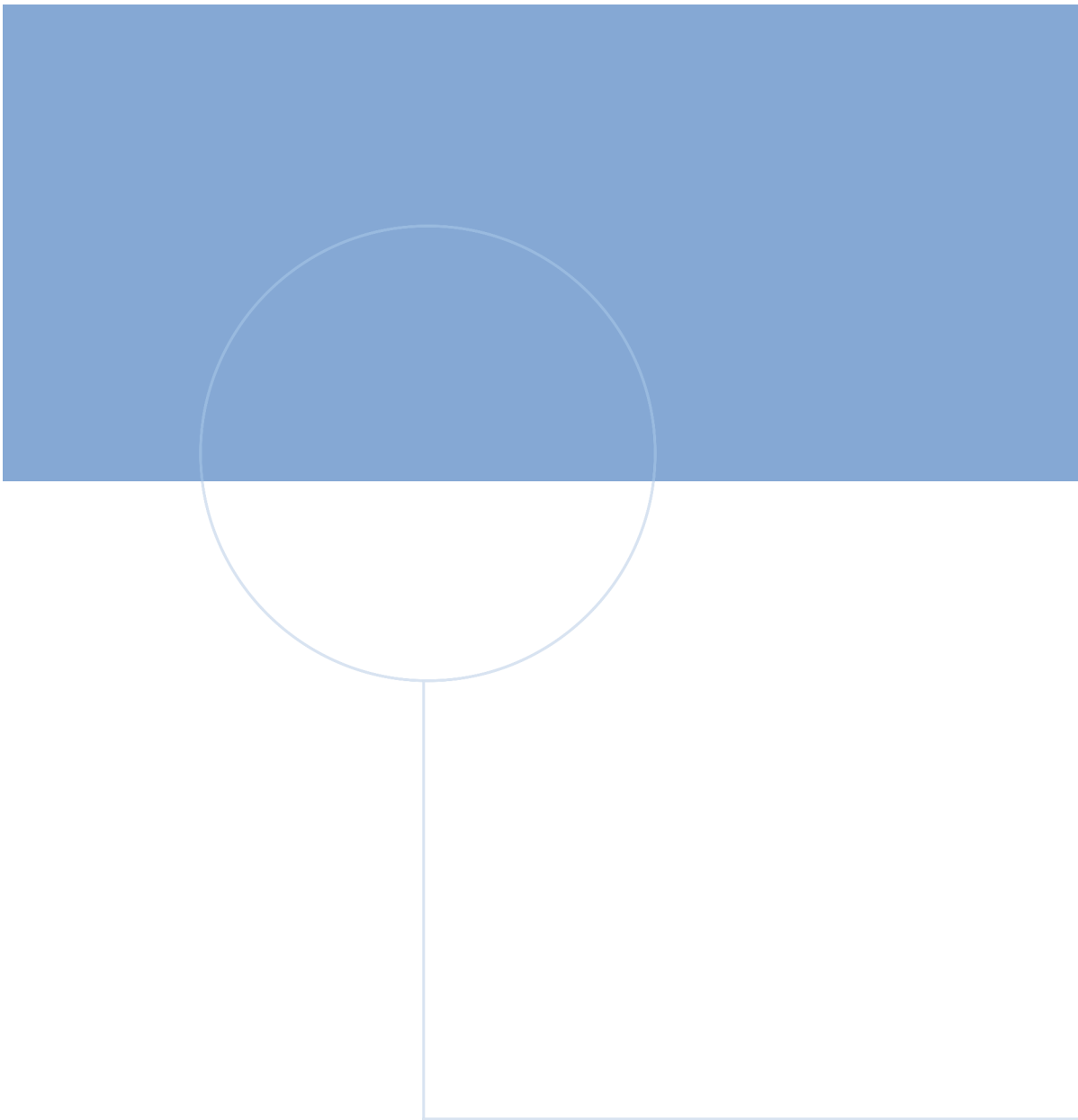
- Ask Kelly if she has any recommended papers or things to include in the related works
- Discuss the structure of related work and problems(relevancy and hybrid models)
- Discuss how the models should be evaluated
- Discuss the structure of the "Evaluation" chapter
- Discuss the use of point adjust in the research paper on RePAD2
- Discuss how detection's should be evaluated (Should only the specific labels be deemed anomalies?)

Notes and further work

may gods grace be bestowed upon his humble soldiers

- Ameen starts in a new position, he might no longer be involved in the project as a supervisor.
- Do not tamper with the datasets like what was done in the research paper on RePAD2. The reason being that they got feedback that they tampered with the dataset and that the labeling should be respected.
- When evaluating we should use varied datasets, this is to better scope the capabilities of the models. If we end up using only NAB datasets we should argue for why we do this in the paper to avoid sensors asking us, at the very least we should be prepared for the question.
- The evaluation part should include: explanation of how the evaluation was done and what was taken into consideration, which metrics was used and which and why we the evaluation included the 6 versions that it does.
- The evaluations on the pre-discussed datasets should be done before the next meeting.
- Incremental learning needs not a separate section in the chapter about related works, and would be better to introduce/cite when introducing the concept later on in the report.

- When structuring the sections of related work, hybrid models could be introduced in the section that the model leans heaviest on. This should be done later in the section.



NTNU

Norwegian University of
Science and Technology