
# CADLAE: Configurable Anomaly Detection, Localisation and Explanation Framework for Cyber Physical Systems

Author: **Cameron J. Looney**
Supervisor: **Prof. Gregory Provan**

A thesis submitted in partial fulfillment
of the requirements for the degree of
Bachelors of Science
Mathematics and Computer Science

at
University College Cork
2023

# CADLAE: Configurable Anomaly Detection, Localisation and Explanation Framework for Cyber Physical Systems

**Cameron J. Looney**

## Abstract

This thesis paper addresses the issue of anomaly detection in cyber-physical systems (CPS), which are systems that combine physical and computational components to monitor and control physical processes. In CPS, anomaly detection is a crucial task, as it can help prevent system failures, ensure safety, and optimise performance. Anomalies in CPS can arise due to a range of reasons, including sensor malfunctions, faults in communication channels, and cybersecurity attacks.

To address the problem of anomaly detection, the proposed method in this thesis is an Seq2Seq LSTM Autoencoder model. An autoencoder is an unsupervised learning method that learns to encode data into a lower-dimensional space and then reconstruct it. In this approach, the encoder encodes the input data, and the decoder reconstructs it back to its original form. The LSTM (Long Short-Term Memory) component of the model allows for the detection of temporal dependencies, which are essential for detecting anomalies in time series data, as in CPS.

Most research in anomaly detection in CPS focuses solely on detecting anomalies, and few attempts have been made to localise them. This thesis, however, aims to go beyond standard anomaly detection by attempting to localise the anomalies. Localisation is essential in CPS as it can help identify the root cause of the anomaly, which is crucial for addressing the underlying issue.

The proposed method uses the feature-wise error generated from model predictions to localise anomalies. The feature-wise error measures the difference between the actual value of a feature and its predicted value by the model. The proposed method applies several techniques to this error to localise the anomaly. These include PCA (Principal Component Analysis) localisation and thresholding localisation.

PCA localisation is a technique that attempts to reduce the dimensionality of the feature-wise error space by projecting it onto a lower-dimensional subspace. This technique is applied to subgraphs, which are groups of highly correlated features that are used to detect and localise anomalies. The purpose of this technique is to identify the features that are contributing the most to the anomaly.

Thresholding localisation is another technique used in the proposed method, which involves setting a threshold for the feature-wise error. The threshold is set based on the maximum feature-wise error in the training set. Any feature whose error exceeds this threshold is considered anomalous and is localised.

Localising anomalies to a group of features rather than a single one is essential in CPS as features are often highly interdependent. Identifying a single anomalous feature may not be enough to determine the root cause of the anomaly, as it may be influenced by other features in the system. The proposed method attempts to identify a group of highly correlated features that are grouped together using their correlation to form disjoint

subgraphs.

The localised anomalies are explained using Gradient Boosting Machines (GBMs), which are powerful machine learning models that combine multiple decision trees to make accurate predictions. The GBM model leverages the predictions from the unsupervised model as labels to train the supervised explanation model, allowing for better prediction accuracy.

One of the key benefits of the GBM model is that it generates if-then rules that can be easily interpreted and acted upon by humans. Additionally, the model enables the calculation of the importance of feature selection and contribution, which provides insight into how the model is making its predictions. By expressing the rules in terms of input features and predicted labels, these insights can be transformed into actionable insights that can be used to prevent further anomalies.

Ultimately, by using the knowledge of the system, the generated rules can be translated into actionable feedback that can be used to improve the system and prevent future anomalies. This approach offers a valuable solution for addressing localized anomalies and improving system performance overall.

An alternative approach to explaining anomalies in cyber-physical systems was explored through the use of Bayesian networks. Bayesian networks utilised a directed acyclic graph (DAG) to represent the joint probability distribution of a set of random variables and their dependencies. In the context of cyber-physical systems, these models were used to analyse the system's behaviour and model the relationships between different components.

To compute the posterior probability of the causes of an anomaly using Bayesian networks, the likelihood, prior, and evidence terms were calculated based on a set of observed variables and the occurrence of the anomaly. Once the posterior probability of the causes had been determined, the Bayesian network could be used to identify the factors that contributed to the anomaly by computing the posterior probability of each variable in the network. This enabled the ranking of the factors that contributed to the anomaly and identification of the most likely causes.

When building a Bayesian network for a cyber-physical system, careful consideration of the system's topology and relevant variables was required to accurately reflect the causal relationships between the variables. A systematic and rigorous approach was necessary to ensure the resulting Bayesian network accurately modelled the system's behaviour.

To evaluate the proposed method's performance, it is compared to different models and datasets. This is done to avoid bias as the Tennessee Eastman data used in this study was generated by the author. By benchmarking the proposed method against other models and datasets, this study aims to provide valuable insights into the field of anomaly detection in CPS.

In summary, this thesis proposes a novel approach for detecting and localising anomalies in CPS using an Seq2Seq LSTM Autoencoder model. The proposed method goes beyond standard anomaly detection by localising anomalies to a group of highly related features, which is crucial in CPS, where features are highly interdependent. The proposed method is evaluated against different models and datasets to provide valuable insights for improving system efficiency and safety.

# Declaration

I hereby declare that the thesis entitled CADLAE: Configurable Anomaly Detection, Localisation and Explanation Framework for Cyber Physical Systems, submitted to University College Cork, is my original work and has not been submitted, in whole or in part, for any other academic award or degree.

I further declare that all the sources used in the research for this thesis have been appropriately cited and acknowledged. The ideas, arguments, and conclusions presented in this thesis are entirely my own, except where otherwise stated.

I acknowledge that any act of academic dishonesty, including plagiarism, is a serious offense that undermines the integrity of the academic community. Therefore, I have taken all necessary steps to ensure that the work presented in this thesis is entirely original and free from any form of academic dishonesty.

I understand that any violation of academic integrity or dishonesty may result in severe consequences, including the revocation of the degree or academic award conferred upon me by the institution.

# Acknowledgements

I am deeply grateful to Prof. Gregory Provan, whose expert guidance, invaluable feedback, and unwavering support have been instrumental in bringing this thesis to fruition. His mentorship has been a constant source of inspiration and motivation, and I feel privileged to have had the opportunity to work under his supervision.

I would also like to extend my appreciation to Dr. James Doherty, who kindly agreed to serve as my second reader for this thesis. I am truly grateful for his time and effort.

Furthermore, I would like to acknowledge the many staff members at University College Cork who have supported me throughout my academic journey.

Finally, I would like to express my gratitude to my family and friends, whose love and encouragement have sustained me throughout this process.

Thank you all for your support and guidance.

# Contents

**1 Introduction** **11**
1.1 Objective . . . 11
1.2 Overview . . . 11
1.3 Definition of Cyber Physical Systems . . . 12
1.4 Anomaly Detection in Cyber Physical Systems . . . 13
1.5 Anomaly Detection using Deep Learning . . . 13
1.6 Challenges . . . 14
1.6.1 High Dimensionality . . . 14
1.6.2 Complex Data Interdependencies . . . 15
1.6.3 Lack of Labelled Data . . . 15

**2 Literature Review** **16**
2.1 Anomaly detection in Cyber Physical Systems . . . 16
2.1.1 Resources for Reader . . . 16
2.1.2 Importance of anomaly detection . . . 17
2.1.3 Types of Anomalies . . . 19
2.2 Model-Based vs Data Driven Approaches . . . 20
2.2.1 Model-Based . . . 20
2.2.2 Data-Driven . . . 21
2.2.3 Conclusion . . . 22
2.3 Deep learning for anomaly detection . . . 22
2.3.1 Overview . . . 22
2.3.2 Autoencoders . . . 23
2.3.3 Recurrent Neural Network . . . 25
2.3.4 Generative Adversarial Networks . . . 27
2.3.5 Hybrid Models . . . 28
2.4 Explaining Anomalies . . . 30
2.4.1 Causal Bayesian Network . . . 30
2.4.2 Tree Based Methods . . . 31
2.5 Anomaly Localization . . . 32
2.6 Conclusion . . . 33

**3 Related Work** **35**

**4 Tennessee Eastman Process: Key Terminology and Context 39**
4.1 The Tennessee Eastman Process . . . 39
4.2 What are Features in Tennessee Eastman? . . . 40
4.2.1 Tennessee Eastman Feature Key . . . 41
4.3 Anomalies in Tennessee Eastman . . . 42
4.3.1 Types of Faults . . . 43
4.4 Localising Anomalies . . . 46
4.5 Explaining Anomalies . . . 47
4.6 Conclusion . . . 48

**5 Proposed Methodology 49**
5.1 Data Generation . . . 49
5.2 Data Preprocessing . . . 50
5.3 Seq2Seq LSTM Encoder Decoder . . . 51
5.4 Optimizing Threshold . . . 53
5.5 Localisation . . . 53
5.5.1 Principal Component Localisation . . . 54
5.5.2 Threshold Localisation . . . 55
5.5.3 Correlation Subgraph Localisation . . . 56
5.6 Explanation . . . 57
5.6.1 Tree Based Explanation . . . 57
5.6.2 Causal Bayesian Network . . . 60
5.6.3 Causal Explanation vs Correlation Localisation . . . 63

**6 Evaluation and Results 64**
6.1 Comparison Models . . . 64
6.1.1 Probabilistic Models . . . 64
6.1.2 Change Point . . . 65
6.1.3 Proximity Based . . . 66
6.1.4 Graph Based . . . 67
6.1.5 Deep Learning . . . 67
6.2 Evaluation Metrics . . . 69
6.3 Results . . . 72
6.3.1 Single Fault Data - Tennessee Eastman . . . 72
6.3.2 Multi Fault Data - Tennessee Eastman . . . 74
6.3.3 Additional Datasets . . . 77
6.3.4 Localisation . . . 80
6.4 Hyper Parameter Tuning . . . 81
6.4.1 Loss Function Comparison . . . 82

**7 Discussion 85**
7.1 Model Performance . . . 85
7.1.1 Tennessee Eastman Single Fault . . . 85
7.1.2 Tennessee Eastman Multi Fault . . . 86
7.1.3 Benchmark Datasets . . . 86
7.1.4 Overall performance . . . 86

7.2 Limitations of Study . . . . . 87
7.2.1 Limitations with Data Used . . . . . 87
7.2.2 Limitations in Anomaly Detection Approach . . . . . 88
7.2.3 Limitations in Localisation . . . . . 88
7.2.4 Limitations in Explainability . . . . . 90
7.3 Deep Learning vs Traditional Techniques . . . . . 91
7.4 Challenge with Limited Data and Privacy Concerns . . . . . 92
7.5 Causality vs Correlation . . . . . 93
7.6 Balancing Accuracy and Interpretability . . . . . 93
7.7 Importance of Interpretability and Explainability . . . . . 95
7.8 Potential of Causal Models . . . . . 95
7.8.1 Promising Techniques . . . . . 95
7.8.2 Advantanges and Disadvantages of Causal Models . . . . . 96
7.9 Future Work . . . . . 97
7.9.1 Focus on Explainability . . . . . 98
7.9.2 Exploring Other Detection Techniques . . . . . 98

**8 Conclusion** **99**

**A Appendix** **100**

# List of Figures

2.1 Example of Power Grid Anomaly Detection Architecture [52] . . . . . 17
2.2 LSTM Memory Cell Architecture [189] . . . . . 26

4.1 Tennessee Eastman Process [64] . . . . . 39
4.2 Tennessee Eastman Feature Key [32] . . . . . 40
4.3 Tennessee Eastman Process Manipulated Variables[64] . . . . . 42
4.4 Tennessee Eastman Process Fault Examples . . . . . 43
4.5 Step Fault Example . . . . . 44
4.6 Random Variation Faults . . . . . 45
4.7 Tennessee Eastman Process Fault Examples . . . . . 45
4.8 Tennessee Eastman Process Fault Examples . . . . . 46
4.9 Tennessee Eastman Process Subsystems [119] . . . . . 47

5.1 Normal vs Anomalous Data in sensor XMEAS(38) for Fault F . . . . . 50
5.2 CADLAE Architecture . . . . . 51
5.3 CADLAE Architecture: LSTMED . . . . . 52
5.4 Algorithm to Calculate Youden's J-statistic . . . . . 53
5.5 Example of PCA Localisation on Reconstruction Error . . . . . 54
5.6 Result of localisation using Demonstration (PCA) . . . . . 55
5.7 Result of localisation using Demonstration (Thresholding) . . . . . 55
5.8 CADLAE Architecture: Feature Wise / Subgraph Localisation . . . . . 56
5.9 Result of localisation using Demonstration (Subgraph) . . . . . 56
5.10 Correlation Disconnected Subgraphs . . . . . 57
5.11 CADLAE Architecture: GBM Explainer . . . . . 58
5.13 GBM Decision Boundary . . . . . 58
5.14 Automatically Generated Human Readable Feedback . . . . . 58
5.12 Gradient Boosting Machine Global Decision Tree . . . . . 59
5.15 CADLAE Architecture: Bayesian Network . . . . . 61
5.16 Tennessee Eastman Bayesian Network . . . . . 62
5.17 Conditional Probabilities for Explaining TEP Example . . . . . 62
5.18 Subsystem Effected in Example (Figure 4.9) . . . . . 63

6.1 ROC-AUC Plot for Fault F . . . . . 71
6.2 Demonstration of Single Fault . . . . . 72
6.3 Visualisation of Single Fault Results . . . . . 74
6.4 Proposed Model Confusion Matrix - Single Fault . . . . . 75

6.5 Multi Fault TEP Example . . . . . 76
6.6 Visualisation of Multi Fault Results . . . . . 77
6.7 Proposed Model Charts for Multi Fault Data . . . . . 78
6.8 Subgraphs displayed in Table 6.5 . . . . . 81

A.1 Proposed Model Confusion Matrix - Single Fault . . . . . 101
A.2 Spearman Rank Correlation Coefficient for TEP . . . . . 102
A.3 Tennessee Eastman - Process Variables . . . . . 103

# List of Tables

4.1 Table of Faults in Simulated Data . . . . . 43

6.1 Results Table for Single Fault Data . . . . . 73
6.2 Results Table for Multi Fault Data . . . . . 76
6.3 Benchmark Datasets Results . . . . . 79
6.4 Benchmark Datasets Overview . . . . . 80
6.5 Fault Localisation Results . . . . . 80
6.6 Results Table for Loss Functions . . . . . 84

# Chapter 1

# Introduction

## 1.1 Objective

The objective of this research is to develop and demonstrate a proof-of-concept for a unified framework that can accurately detect, localise, and explain anomalies in cyber-physical systems (CPS) in an interpretable manner. The proposed framework is designed to address the significant challenges associated with detecting and localising anomalies in CPS, which often involve complex and dynamic interactions between physical and cyber components.

To achieve this objective, the study will focus on a simulated CPS, specifically the Tennessee Eastman process, which is a well-established benchmark dataset for anomaly detection in CPSs. The proposed framework will be evaluated on this dataset and compared with other existing models to demonstrate its effectiveness.

The proposed approach will leverage the strengths of various anomaly detection and localisation techniques to develop a unified framework that can accurately detect and localise anomalies in CPSs.

Moreover, the framework will also provide interpretable explanations for detected anomalies, enabling system operators to understand the underlying causes of the anomalies and take appropriate corrective actions. This feature is essential to ensure that the framework is not only accurate but also transparent and explainable, which is crucial for building trust in the system and promoting wider adoption.

## 1.2 Overview

Anomaly Detection and Fault Detection are both critical tasks in the field of Cyber Physical Systems which will be referred to as CPS throughout this paper [113]. CPS are usually physical systems that are embedded with sensors and computing capabilities allowing them to collect data from the physical environment [11] and to interact with said physical environment. In the modern age CPS are becoming an essential and constant presence. Examples of CPS include smart grids, autonomous vehicles and industrial control systems such as a water plant [118]. As CPS are reliant on consistent and accurate data to function effectively, they are extremely vulnerable to both anomalies and faults that can comprise this data and thus compromise the functionality and reliability of the

entire system [121].

As a result of this, effective anomaly detection is one of the most critical and vital tasks in the CPS [113] space to ensure the continued operation and safety of systems that are becoming increasingly inter-wined with daily life.

Deep learning [128] is a type of machine learning that involves the use of neural networks to learn complex relationships from large amounts of data [101]. It has been widely used in various fields, including computer vision, natural language processing, and anomaly detection. In recent years, there has been growing interest in using deep learning for anomaly detection in CPS [113].

The Tennessee Eastman testbed is a CPS benchmark for studying the detection and diagnosis of anomalies [28]. It simulates the behavior of a chemical process and can generate various types of anomalies, including normal operations, and process faults.

The aim of this research is to investigate the use of deep learning for anomaly detection in CPS, with a focus on the Tennessee Eastman testbed. The research will propose a deep learning-based approach for detecting anomalies in the Tennessee Eastman testbed and evaluate its performance in comparison to existing methods.

In Section 2, a comprehensive literature review will be conducted to review and discuss relevant studies on anomaly detection in CPS and deep learning. Section 3 will provide details on the proposed methodology which includes the use of the Tennessee Eastman testbed and a deep learning-based approach for identifying anomalies in CPS. The evaluation and results of the proposed method will be reported in Section 4. In Section 5, the implications of the findings will be examined and recommendations for additional research will be provided. Overall, this study aims to provide a comprehensive examination of the use of deep learning for anomaly detection in CPS and its effectiveness in identifying anomalies in real-world industrial systems.

## 1.3 Definition of Cyber Physical Systems

A class of complex systems known as "Cyber Physical Systems" combines physical, networking, and computational activities. They are made up of two basic parts: the physical parts, which are the system's physical processes and components, and the Cyber parts, which are the system's computing processes and components [42]. The physical parts of CPS can be made up of a variety of components, including embedded controllers, robotics, sensors, and actuators[42]. Sensors are used to measure and detect outside events like pressure, temperature, and motion. Actuators, which include motors, valves, and solenoids, are used to regulate and modify the physical world. The system's computational power is provided by embedded controllers, which are integrated into the physical environment [103].

The Cyber components of CPS are composed of various elements, such as embedded processors, embedded software, and communication networks [103]. Embedded processors are specialized microprocessors used to provide the computing power required to control the physical components. Embedded software is the software code that runs on the embedded processors and is responsible for controlling the physical components of the system [102]. Communication networks are used to connect the physical components to one another and to the outer world, allowing for the exchange of data and commands.

CPS can also include other components, such as system-level software, middleware, and user interfaces. System-level software is a layer of software that provides the overall control and coordination of the system [163]. Middleware is software that provides services to the system, such as communication and data processing [63]. Finally, user interfaces are components that allow users to interact with the system, such as graphical user interfaces and voice recognition systems [127][129].

## 1.4 Anomaly Detection in Cyber Physical Systems

Anomaly detection is an important tool for defending against cyberattacks in CPS. The rapid expansion of CPS has led to an increase in attacks against them [170][92]. Anomaly detection is essential for detecting and preventing these attacks [36]. Anomaly detection is a process of detecting and identifying unusual patterns or events in data, which can indicate malicious activity. Anomaly detection can be used to detect a variety of malicious activities, including denial of service attacks, reconnaissance activities malicious code execution [134].

CPS have become increasingly connected and interdependent, making them highly vulnerable to cyberattacks [31]. As the number of connected devices increases, the complexity of the system increases, making it difficult to detect and respond to cyberattacks in a timely manner. Anomaly detection can provide an effective solution to this problem by detecting anomalous behavior in the network and alerting administrators of potential threats.

The security of CPS can greatly benefit from anomaly detection systems. These technologies can assist in recovering from and minimising the damage caused by cyberattacks and lowering the risk of faults by spotting and reacting to hostile actions in real-time. Anomaly detection systems can also offer network visibility, giving administrators knowledge of potential threats and weaknesses. Administrators may be better able to see malicious activities and take the proper action with this network visibility.

## 1.5 Anomaly Detection using Deep Learning

Deep learning is a subset of machine learning that has gained significant popularity in recent years due to its ability to learn and generalize from large amounts of data. These networks are composed of many interconnected nodes, or neurons, which process input data and generate output predictions. Deep learning algorithms differ from traditional machine learning methods in that they use multiple layers of neurons to learn hierarchical representations of the data, allowing them to learn more complex and abstract patterns. [137]

Autoencoders are a type of deep learning algorithm that are often employed for anomaly detection. They are a type of neural network designed to reconstruct the input data from a lower-dimensional representation [34]. Autoencoders are trained to minimise the reconstruction error of the input data, and are then used to identify anomalies by detecting input data that cannot be accurately reconstructed. This approach has been shown to be effective for anomaly detection in a variety of applications, including fraud

detection in financial transactions [156] and identification of network intrusions in cybersecurity [120].

Another deep learning approach to anomaly detection is the use of generative adversarial networks (GANs), which are a type of neural network composed of two competing networks: a generator network that produces synthetic data, and a discriminator network that attempts to distinguish between real and synthetic data [37]. GANs are trained by optimising the competing objectives of the two networks, and have been shown to be effective at modeling complex data distributions. In the context of anomaly detection, GANs can be used to learn the normal behavior of the data, and then identify anomalies as data points that are not generated by the generator network. This approach has been applied to a range of applications, including detection of rare events in time series data [104] and identification of anomalous images in large datasets [87].

In general, deep learning techniques have shown tremendous promise for anomaly detection and have been used for a variety of purposes. The effectiveness and resilience of deep learning for anomaly detection still need to be improved, despite the fact that these methods have produced encouraging results.

## 1.6 Challenges

### 1.6.1 High Dimensionality

The enormous dimensionality of the data is one of the main obstacles to anomaly detection in cyber-physical systems, as demonstrated in [7]. Data that has a lot of features or variables is referred to as high-dimensional data, which can make it challenging to effectively identify the underlying trends and discern between normal and abnormal behavior. This is because it becomes more challenging to find patterns and links between the variables as the number of dimensions in the data grows exponentially with the number of features.

The large dimensionality of data in cyber-physical systems can be caused by a number of variables. The enormous number of sensors and other data sources that are frequently employed to monitor these systems is one of the factors. A power grid, for instance, might contain hundreds or even thousands of sensors dispersed across the system, each of which produces a time series of measurements [4]. Due to the fact that each sensor produces a unique data series, the high number of sensors results in a high-dimensional data space.

The duration of the time series is another element that may contribute to the high dimensionality of data in cyber-physical systems. These systems frequently have very long time series data with hundreds or even millions of data points. As a result, the data may become more dimensional as each data point may represent a distinct feature.

Cyber physical systems' large data dimensionality might make it challenging to effectively model the data and spot unusual behavior. One-class SVMs and kernel density estimation are two common traditional approaches for anomaly identification that may struggle to handle high-dimensional data and may not be reliable enough to effectively capture the complex patterns and relationships in the data.

### 1.6.2 Complex Data Interdependencies

The intricate relationships between the data series pose a significant obstacle to anomaly identification in cyber-physical systems. This speaks to the complex interrelationships among the various factors, which can make it challenging to effectively analyze the data and spot abnormal activity.

Consider a water treatment facility that tracks a variety of variables, such as pH, temperature, and chlorine levels, to provide an example of this problem. These variables are frequently interconnected, meaning that changes in one variable may have an impact on the values of other variables. For instance, alterations in the pH of the water can have an impact on how soluble chlorine is, which in turn can have an impact on how much chlorine is present in the water. [117]

In this case, it could be necessary to take into consideration the intricate connections between the various factors in order to spot anomalies in the data. Traditional approaches to anomaly detection might not be adequate to fully understand these relationships, leading to false alarms or missing actual anomalies.

Researchers have created a number of techniques that take into consideration the complicated interdependencies between various data series to address this difficulty. A graph neural network, which is a type of neural network created to function on graph-structured data that is capable of capturing the detailed interactions between various nodes in the graph, is used in [107]. A Bayesian network, a graphical model that depicts the probability correlations between various variables and can be used to describe complicated systems, is utilized in [94].

Both of these methods have produced encouraging results for detecting anomalies in cyber-physical systems, but additional study is still required to enhance their effectiveness and expand the range of systems to which they may be applied.

### 1.6.3 Lack of Labelled Data

The majority of the time, detecting anomalies involves unsupervised learning, which means that the training data does not contain labels indicating whether a specific sample is normal or abnormal. The model must therefore learn to differentiate between typical and abnormal behavior based on the data itself, which might be difficult if the data is very volatile or comprises a sparse collection of anomalous examples.

The absence of labeled data in cyber-physical systems can be attributed to a number of issues. One aspect is the complexity of the systems, which might make it challenging to identify the data appropriately. The behavior of a power grid, for instance, can be influenced by a variety of circumstances, such as weather conditions and equipment failures, making it challenging to evaluate whether a given reading is normal or abnormal.

The cost and duration involved in labeling the data is another aspect. Data labeling frequently calls for the knowledge of subject matter experts, who could be hard to come by or perhaps too busy to classify a lot of data. Additionally, categorizing the data can take a while, particularly for systems with a lot of data series or a lengthy time horizon.

# Chapter 2

# Literature Review

## 2.1 Anomaly detection in Cyber Physical Systems

### 2.1.1 Resources for Reader

In preparation for this literature review on anomaly detection in cyber physical systems, it is imperative to acknowledge the existence of several comprehensive surveys and reviews in this field. In order to provide the reader with a well-rounded understanding of the current state-of-the-art research in anomaly detection, this section will present a summary of the most relevant and recent sources available. These resources will serve as a foundation for comprehending the advancements made in the field and provide insight into the challenges and opportunities for future research.

The paper titled "Deep Learning for Anomaly Detection: A Review" [131] authored by Pang et al. provides a comprehensive survey of the research in the area of deep anomaly detection. Anomaly detection, also known as outlier detection or novelty detection, is a long-standing research area in various research communities. The authors aim to provide a comprehensive overview of the advancements in this area in recent years, with a particular focus on deep learning-based approaches.

The authors provide a taxonomy of the methods in deep anomaly detection, covering advancements in three high-level categories and 11 fine-grained categories. They review the key intuitions, objective functions, underlying assumptions, advantages, and disadvantages of the methods and discuss how they address the challenges posed by the problem. The authors also provide a discussion of the future opportunities and new perspectives on addressing the challenges in the field.

The authors provide a rigorous discussion of the methods, including their objective functions and underlying assumptions, as well as their advantages and disadvantages. The objective functions of the methods vary, with some methods focusing on reconstructing the data and others focusing on classifying anomalies. The underlying assumptions of the methods also vary, with some methods assuming that the normal data is well-behaved and others assuming that the anomalies are rare and difficult to model.

In "Deep Learning-Based Anomaly Detection in Cyber-Physical Systems: Progress and Opportunities," [114] Yuan Luo et al. describe the current state of deep learning-based anomaly detection (DLAD) methods for cyber-physical systems. They aim to understand the essential properties of these methods and highlight new characteristics and designs in

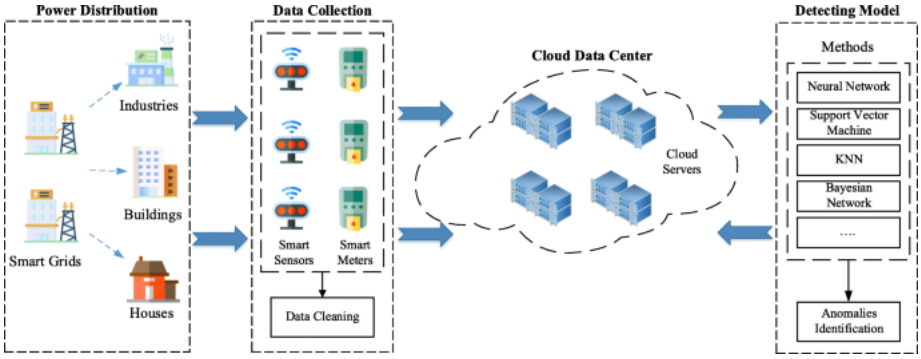



Figure 2.1: Example of Power Grid Anomaly Detection Architecture [52]

each CPS domain.

The authors begin by discussing the importance of anomaly detection in ensuring the security of CPSs, and why conventional anomaly detection methods are becoming increasingly inadequate for this task. They then present a taxonomy of DLAD methods in terms of the type of anomalies, strategies, implementation, and evaluation metrics. This taxonomy is used to identify new characteristics and designs in each CPS domain and to discuss the limitations and open problems of these methods.

The authors also provide insights into the selection of proper DLAD methods in practice by experimentally exploring the characteristics of typical neural models, the workflow of DLAD methods, and the running performance of DL models. The results of these experiments are then used to discuss the deficiencies of DL approaches, as well as to provide possible directions for improving DLAD methods and motivate future research.

### 2.1.2 Importance of anomaly detection

Any Cyber physical system must include anomaly detection since it includes the identification of unusual or unexpected behavior within the system. It is possible to prevent failures and preserve the safe and dependable operation of the system by identifying and resolving any anomalies that may be caused by this unusual behavior. As proposed in [192], anomaly detection is a key tool for ensuring the reliable and efficient operation of modern power grids. Anomalous behaviors, such as unusual consumption patterns of the users, faulty grid infrastructures, outages, external cyberattacks, or energy fraud, can be identified and prevented with the use of anomaly detection methods [35].

The ability of anomaly detection to spot patterns in the system's data is a crucial feature. Numerous techniques, including statistical analysis, machine learning algorithms, and rule-based systems, can be used to do this. For instance, a typical strategy is to use a probability distribution to represent the expected behavior of the system, and then to identify any departures from this distribution as probable anomalies. In [142] the use of anomaly detection techniques to identify potential flaws in Clinical Decision Support (CDS) systems was demonstrated. Six models, including Poisson Changepoint detection, ARIMA, Hierarchical Divisive Changepoint Model, Bayesian Changepoint Model, Seasonal Hybrid Extreme Studentized Deviate model (S-H-ESD), and E-Divisive with Median, were employed to detect point and Changepoint anomalies in CDS alert firing

data. In order to identify the beginning and end dates of anomalies, these models were applied to four time series with known flaws. The findings demonstrated that, despite varied degrees of false alarm rates and detection delays, that statistical models could be successful in identifying anomalies.

Anomaly detection can also assist in locating the source of a systemic problem. It is possible to identify the underlying reason of an anomaly by examining the data and the system's behavior, which enables a more focused and efficient reaction. This can assist in reducing the chance of a similar problem reoccurring in the system. This is a key focus of this paper as the research in this critical area is minimal despite its importance. In [159], the authors present a random matrix theory-based approach for early anomaly detection and localisation in distribution networks. The approach leverages the similarities of the data collected from multiple monitoring devices and is capable of detecting and localising anomalies at an early stage. It is robust to random disturbance and measurement error, and a data dimensionality increasing algorithm is designed for analysing the monitoring data from low observability feeders more accurately. The benefits of this approach are clear. By localising anomalies at an early stage, operators can take action quickly to prevent further spread of the anomaly and minimise the potential damage that it may cause.

[168] introduces a semi-supervised approach for detecting and localising cyberattacks in water distribution systems. This approach utilises maximum canonical correlation analysis (MCCA) to reduce the dimensionality of the problem and support vector data description (SVDD) to classify anomalous samples without needing labeled attacks in the training dataset. The method was tested on two case studies and various datasets, demonstrating consistently high performance in detecting and localising cyberattacks. This research highlights the importance of localizing anomalies, as it allows for a more efficient response to cyberattacks and can help to prevent further damage to the system. Furthermore, the development of a semi-supervised approach is a major benefit, as it is more applicable to real-world applications which lack labeled-attacks datasets.

Failure to recognize anomalies in a CPS might have serious repercussions. For instance, undiscovered anomalies may cause system failures, which may seriously harm the system and its parts. Undiscovered abnormalities can also endanger human lives in safety-critical systems, such as those utilised in transportation, healthcare, energy or critical infrastucture such as water treatment facilities. The notorious Stuxnet worm demonstrated that many of the security assumptions made about operational environments, technological capabilities, and potential threat risks are far away from reality and present a huge challenge for modern industrial systems. [82] investigates the highly sophisticated aspects of Stuxnet and the impact it has on existing security considerations to pose thoughts on the next generation of Supervisory Control and Data Acquisition (SCADA) systems from a security perspective. It highlights the importance of incorporating risk analysis and security tools into industrial systems in order to detect anomalies and prevent attacks. Additionally, it suggests that security should be integrated into all operational aspects of industrial systems and that industry should invest in upgrading the security of their systems to ensure the safety of their processes and critical infrastructure. This is a key takeaway, as security in the Cyber domain is often times an afterthought. Where investment is made in a reactive position rather than as a preventative measure.

### 2.1.3 Types of Anomalies

#### Temporal Anomalies

Temporal anomalies in CPS are an increasingly relevant research topic due to the rise of the Internet of Things (IoT) and increasing use of embedded systems. The temporal aspects of CPS are critical to its functioning, however, temporal anomalies can cause system malfunctions and lead to erroneous results.

A temporal anomaly occurs when the temporal characteristics of a system are not as expected or when the system does not meet its temporal requirements. Generally, temporal anomalies can be classified into two categories: process anomalies and timing anomalies. Process anomalies occur when the order of events or actions are different from what is expected, while timing anomalies occur when the timing of events or actions deviate from what is expected.

The formal definition of temporal anomalies in CPS can be expressed using a mathematical equation. Let $T_a$ and $T_b$ represent the expected temporal characteristics of two processes in a CPS, and $T_c$ represent the actual temporal characteristics of the processes. Then, the equation for temporal anomalies in CPS can be expressed as:

$$\frac{|T_a - T_b|}{T_c} > 1 \tag{2.1}$$

The absolute difference between two processes' expected temporal characteristics divided by their actual temporal characteristics must be more than one in order for there to be a temporal anomaly. The author of reference [81] suggests a technique for identifying abnormalities in the operation of CPS by studying multidimensional time series. Recurrent GRU neural networks are used in the method to anticipate the values of time series of system data and to find discrepancies between the expected value and the most recent data acquired from sensors and actuators. By comparing the anticipated values with the actual values, the neural network is utilised to detect any abnormalities or cyber threats. It is trained on the data acquired from the CPS. The results of experimental studies demonstrated the effectiveness of the proposed solution.

#### Statistical Anomalies

In Cyber physical systems, statistical anomalies can have huge repercussions since they can reveal that the system is experiencing an unanticipated or abnormal event. Understanding the predicted statistical features of the data generated by the system is crucial for both detecting and analyzing these anomalies.

To examine statistical abnormalities in CPS data, a variety of mathematical methods and tools can be applied. One typical approach is to utilize statistical distributions and probability theory to explain the anticipated behavior of the system. One popular distribution that can be used to simulate the behavior of various CPS systems is the normal distribution.

Data that is "out of distribution," or outside the range of the data used for training, can produce significant errors and jeopardize safety. Inductive conformal prediction and anomaly detection are used to create a method for effectively identifying out-of-distribution data in CPS control systems in this study [26]. The suggested method makes use of deep

support vector data descriptions and variational autoencoders to create models that can quickly compute how well new inputs conform to the training set. This enables the real-time detection of out-of-distribution data. An enhanced emergency braking system and a self-driving end-to-end controller developed in a self-driving car simulator are used to illustrate the concept. With execution speeds similar to those of the original machine learning components, the simulation results demonstrate a very low proportion of false positives and detection delays.

Another important tool is statistical inference, which allows us to make predictions about the behavior of the system based on a sample of the data. This can be done using methods such as maximum likelihood estimation or Bayesian inference.

In the paper [94] the authors propose a Bayesian network approach for detecting and isolating anomalies in CPS using unlabeled data. The approach involves transforming the cyber and physical data to make it suitable for learning the Bayesian network structure and parameters. The authors also present scalable algorithms for detecting different types of anomalies and isolating their root causes using the Bayesian network. The performance of the proposed approach is evaluated using an unlabeled dataset consisting of anomalies due to both cyber attacks and physical faults in a commercial building system. The use of statistical inference techniques such as the one used here allow for more robust localization of anomalies and thus have immediate application in real world scenarios.

#### Functional Anomalies

It's critical to establish reliable procedures for identifying and correcting functional anomalies in CPS because they can have detrimental effects. Both model-based and data-driven approaches can be successful at identifying functional anomalies, but it's necessary to take into account their respective shortcomings and potential difficulties. Furthermore, choosing the best course of action to alleviate functional anomalies requires an awareness of their underlying causes. In the section that follows, both model-based and data-driven strategies will be examined in greater detail.

## 2.2 Model-Based vs Data Driven Approaches

### 2.2.1 Model-Based

Model-based methods use a mathematical representation of the system to forecast its behavior and identify departures from the expected behavior. These models may be based on engineering concepts, physical laws, or other sources of system knowledge. The ability to provide a greater knowledge of the underlying reasons of the abnormalities is one benefit of model-based techniques. A model-based method, for instance, may be able to pinpoint the precise part that is generating an anomaly in a power system, such as a malfunctioning transformer or a damaged power line.

In discrete event systems, the model-based anomaly detection method is implemented in the publication cited as [88]. To define the expected behavior of the system, a behavior model is used. Model formation is the process of building the model, which is frequently done manually by human engineers. After the model has been created, the actual work of finding anomalies includes contrasting the system's observed behavior with the expected

behavior predicted by the model. If the observed behavior considerably deviates from the expected behavior, there may be an anomaly present. The authors provide a classification system for anomalies in discrete event systems and a custom behavior model termed the Probabilistic Deterministic Timed-Transition Automaton (PDTTA) to capture the crucial elements of the system for identifying these irregularities. Additionally, they suggest a novel method for detecting anomalies based on traversing the automaton and averaging probabilities, as well as a new learning algorithm for PDTTA. Although the method requires a significant upfront investment to create the model, it clearly has distinct benefits when it comes to explaining anomalies.

Another a more relevant example can be seen in [108]. In this paper, a graphical model-based approach is proposed for detecting anomalies in the operation of an Industrial Control System (ICS) called SWaT (Secure Water Treatment) [117]. The approach involves learning timed automata as a model of normal behavior based on sensor signals, and learning Bayesian networks to discover dependencies between sensors and actuators. The learned models are then used as a one-class classifier for anomaly detection, allowing the detection of irregular behavioral patterns and dependencies. The approach is applied to a dataset collected from SWaT and is shown to outperform other methods such as support vector machines and deep neural networks in terms of precision and run-time. The approach is also interpretable, allowing for the localization of abnormal sensors or actuators.

### 2.2.2 Data-Driven

On the other hand, data-driven techniques rely on data gathered from the system to spot anomalies. These methods often employ deep learning techniques to examine data patterns and spot variations from the norm. Data-driven techniques have the benefit of being able to handle more complicated systems because they do not necessitate prior system knowledge. The fundamental reasons of the anomalies may not be as well understood by data-driven approaches as they are by model-based approaches.

A paradigm for identifying anomalies in the network traffic of cyber-physical systems is proposed in the study cited as [152]. The suggested technique allows for the detection of anomalies without the need for specialized knowledge of the protocols or application being used by learning features from the network traffic's raw byte stream using stacked denoising autoencoders. The authors assert that their method can detect longer-lasting attacks with high precision and recall and is faster and more effective than approaches currently in use that rely on packet parsing. The suggested framework aims to overcome the difficulties in identifying anomalies in CPSs, which are more prone to attacks as a result of their growing interconnectedness and use of exclusive protocols.

In [25], the authors discuss the steps involved in detecting anomalies in CPS for both regression and classification tasks using a data driven approach. They propose the use of deep-radial basis function (RBF) networks, which are conventional deep neural networks (DNNs) with an output RBF layer, as a single network for both controller predictions and anomaly detection in CPSs. The authors design deep-RBF networks using popular DNNs and use the resulting rejection class for detecting physical and data poison adversarial attacks. They show that the deep-RBF networks can effectively detect these attacks with limited resource requirements. The authors also propose a new method for adapting the

deep-RBF networks to regression tasks, which involves the use of a weight sharing scheme. They demonstrate the effectiveness of the deep-RBF networks in detecting anomalies in both classification and regression tasks on several CPS datasets, including one collected from a real autonomous car.

The paper [58] presents a novel unsupervised approach for detecting cyber attacks in cyber-physical systems (CPSs) using a recurrent neural network (RNN) and the cumulative sum (CUSUM) method. The authors propose the use of a long short-term memory RNN to predict a sequence of data and identify anomalies in a replicate of a water treatment plant. The proposed method is able to detect the majority of attacks designed by the research team with low false positive rates and also identifies the sensor that was attacked. The method is validated on the Secure Water Treatment (SWaT) [117] testbed, showing its effectiveness in detecting anomalies in real-world CPSs. The authors also propose a new method for adapting the RNN to regression tasks, which involves using a weight sharing scheme. The proposed method has the advantage of not requiring any abnormal data in the training phase and has the potential to be applied to other CPSs for anomaly detection.

### 2.2.3 Conclusion

Both model-based and data-driven approaches for anomaly identification in CPS have a number of benefits and drawbacks. The ability to provide a greater knowledge of the underlying reasons of the abnormalities, as previously indicated, is one benefit of model-based techniques. Given that humans can examine and comprehend the models, they may also be easier to understand. The precision and thoroughness of the models, as well as the assumptions made about the system, might, however, be a limitation of model-based techniques.

On the other hand, data-driven techniques have the benefit of being able to manage more complicated systems without requiring prior knowledge. Due to the fact that they do not rely on specific system assumptions, they can also be more reliable. Data-driven approaches, however, depend on complicated algorithms that are challenging for people to grasp, thus they may not be as interpretable as model-based approaches. Additionally, they might be more susceptible to noise and data outliers, which could result in false positives or false negatives.

In conclusion, both model-based and data-driven approaches for anomaly detection in CPS have their own benefits and drawbacks. While data-driven approaches are more resilient and can manage more complicated systems, model-based approaches can offer a deeper knowledge of the underlying reasons of the anomalies. The specific requirements and limitations of the application should guide the choice of approach.

## 2.3 Deep learning for anomaly detection

### 2.3.1 Overview

Due to its capacity to simulate intricate patterns and connections in data, the machine learning branch of deep learning has attracted a lot of attention lately. Anomaly detection can be performed using a variety of deep learning models, each with specific capabilities and features.

The recurrent neural network is one kind of deep learning model that is frequently used for anomaly detection (RNN). Because they can handle sequential data, like time series data, in a way that captures dependencies between observations over time, RNNs are especially well-suited for this task. An excellent illustration of this can be seen in [158], where an RNN is trained to spot peculiar stock price trends by treating the prices of a given stock over time as a series of observations.

Another type of deep learning model that has been used for anomaly detection is the autoencoder. Autoencoders are neural networks that are trained to reconstruct their input data by learning an encoding and decoding process. They can be used for anomaly detection by training them on normal data and then using them to detect anomalies in new, unseen data. Anomaly detection with autoencoders typically involves comparing the reconstruction error of the autoencoder for the new data to a predetermined threshold which is considered in [12]. If the reconstruction error exceeds the threshold, it is considered an anomaly.

A different kind of deep learning model that has been used to tackle the issue of anomaly detection is generative adversarial networks (GANs). A generator network and a discriminator network are the two neural networks that make up GANs. As mentioned in [38], the discriminator network is trained to differentiate between real and created data, whilst the generator network is trained to generate data that is comparable to the training data. The discriminator network is used to find abnormalities in new, previously unseen data whereas the generator network is trained on regular data in the context of anomaly detection.

Other types of deep learning models that have been applied to the problem of anomaly detection include convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and self-organizing maps (SOMs). CNNs are particularly well-suited for processing image data and have been used for anomaly detection in a variety of applications, including intrusion detection in computer networks and fault detection in industrial processes [61]. LSTM networks, which are a type of RNN, have also been used for anomaly detection in time series data [189]. SOMs are a type of unsupervised learning algorithm that can be used to identify patterns and relationships in data [90] and have been applied to anomaly detection in a variety of contexts.

In general, the properties of the data and the particular requirements of the application determine which deep learning model should be used for anomaly detection. Each kind of model has its own advantages and disadvantages and may be more or less appropriate for a given task. Therefore, while choosing a deep learning model, it is crucial to carefully analyze the attributes of the data and the objectives of the anomaly detection task.

### 2.3.2 Autoencoders

[12] describes autoencoders as consisting of two parts: an encoder, which maps the input data to a lower-dimensional latent space, and a decoder, which maps the latent representation back to the original input space.

The encoder function can be represented mathematically as follows:

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}; \mathbf{w})$$

where $\mathbf{x}$ is the input data, $\mathbf{z}$ is the latent representation of the data in the latent space,

and $\mathbf{w}$ are the weights of the encoder function. The decoder function can be represented as follows:

$$\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z}; \mathbf{w})$$

where $\hat{\mathbf{x}}$ is the reconstruction of the input data by the autoencoder and $\mathbf{w}$ are the weights of the decoder function.

If the reconstruction error exceeds the threshold, it is considered an anomaly. Mathematically, this can be expressed as follows:

$$\text{Anomaly} = \begin{cases} 1 & \text{if } ||\mathbf{x} - \hat{\mathbf{x}}|| > \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{x}$ is the input data, $\hat{\mathbf{x}}$ is the reconstruction of the input data by the autoencoder, and threshold is a predetermined threshold. Mean squared error (MSE) is a common metric for measuring the reconstruction error of an autoencoder. It is defined as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$$

where $n$ is the number of samples in the input data, $\mathbf{x}_i$ is the $i^{th}$ sample in the input data, and $\hat{\mathbf{x}}_i$ is the reconstruction of the $i^{th}$ sample by the autoencoder.

Mean absolute error (MAE) is another metric for measuring the reconstruction error of an autoencoder. It is defined as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\mathbf{x}_i - \hat{\mathbf{x}}_i|$$

where $n$ is the number of samples in the input data, $\mathbf{x}_i$ is the $i^{th}$ sample in the input data, and $\hat{\mathbf{x}}_i$ is the reconstruction of the $i^{th}$ sample by the autoencoder.

Variational autoencoders (VAEs) are a type of autoencoder that are particularly well-suited for anomaly detection. VAEs are trained to learn a distribution over the input data and can be used to identify unusual patterns or events in data by comparing the likelihood of the new data under the learned distribution to a predetermined threshold. In [8] this is done by comparing the negative log-likelihood of the VAE for the new data to a predetermined threshold. If the negative log-likelihood exceeds the threshold, it is considered an anomaly. Mathematically, this can be expressed as follows:

$$\text{Anomaly} = \begin{cases} 1 & \text{if } -\log p(\mathbf{x}|\mathbf{z}) > \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

where $p(\mathbf{x}|\mathbf{z})$ is the likelihood of the input data $\mathbf{x}$ under the learned distribution and threshold is a predetermined threshold. The negative log-likelihood can be calculated as follows:

$$-\log p(\mathbf{x}|\mathbf{z}) = \text{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] + E_{q(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z})]$$

where $\mathrm{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$ is the Kullback-Leibler divergence between the posterior distribution $q(\mathbf{z}|\mathbf{x})$ and the prior distribution $p(\mathbf{z})$, and $E_{q(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z})]$ is the expected value of the negative log-likelihood of the input data $\mathbf{x}$ under the learned distribution.

The Kullback-Leibler divergence (KL divergence) [80] is a measure of the difference between two probability distributions. In the context of VAEs, it measures the difference between the posterior distribution $q(\mathbf{z}|\mathbf{x})$ and the prior distribution $p(\mathbf{z})$. It is defined as follows:

$$\mathrm{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] = \int q(\mathbf{z}|\mathbf{x}) \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} d\mathbf{z}$$

The posterior distribution $q(\mathbf{z}|\mathbf{x})$ is the distribution over latent variables $\mathbf{z}$ given the input data $\mathbf{x}$. It is typically approximated using an encoder network in a VAE. The prior distribution $p(\mathbf{z})$ is a fixed distribution over the latent variables, such as a standard normal distribution.

The expected value of the negative log-likelihood, $E_{q(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z})]$, is the average negative log-likelihood of the input data $\mathbf{x}$ under the learned distribution, with respect to the posterior distribution $q(\mathbf{z}|\mathbf{x})$. It can be calculated as follows:

$$E_{q(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z})] = \int q(\mathbf{z}|\mathbf{x})(-\log p(\mathbf{x}|\mathbf{z}))d\mathbf{z}$$

In a VAE, the negative log-likelihood of the input data $\mathbf{x}$ under the learned distribution can be calculated by adding the expected value of the negative log-likelihood of the input data under the learned distribution to the Kullback-Leibler divergence between the posterior distribution and the prior distribution. By comparing it to a predefined threshold, this negative log-likelihood can be utilized to find anomalies. Anomaly is defined as the negative log-likelihood above the threshold.

### 2.3.3 Recurrent Neural Network

In the context of anomaly detection, RNNs can be used to identify unusual patterns or events in data by learning a representation of normal behavior and then detecting deviations from this representation. This can be accomplished in a number of ways, such as by training the RNN to reconstruct the input data or by training it to classify normal and anomalous data [173].

[188] implements an autoencoder RNN for anomaly detection tasks. Autoencoders, as discussed above are neural networks that are trained to reconstruct their input data by learning an encoding and decoding process. They can be used for anomaly detection by training them on normal data and then using them to detect anomalies in new, unseen data [188].

Utilizing an RNN as a classifier is an additional strategy for training it to detect anomalies. The RNN must be trained to determine if a particular series of data is typical or abnormal. The RNN can be trained on a labeled dataset that includes both normal and anomalous data using a supervised learning approach, which can be used to accomplish this. A different method is to employ unsupervised learning, where the RNN is trained solely on normal data and then used to categorize new, untouched data as either normal or anomalous.

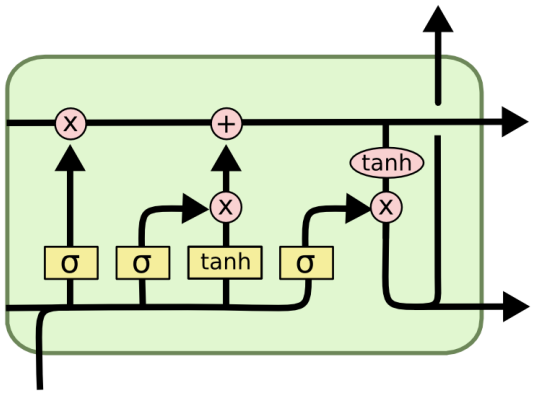



Figure 2.2: LSTM Memory Cell Architecture [189]

In both cases, the goal is to train the RNN to learn a representation of normal behavior that can be used to identify deviations from this behavior in new, unseen data. This can be achieved by minimizing the difference between the output of the RNN and the true label or by minimizing the reconstruction error, depending on the approach being used [13]. For example, in the case of supervised learning, the objective function to be minimized is typically the cross-entropy loss, which is defined as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

where $N$ is the number of samples in the training dataset, $y_i$ is the true label for the $i$th sample, and $\hat{y}_i$ is the predicted label for the $i$th sample.

One key advantage of using RNNs for anomaly detection is their ability to capture long-term dependencies in data. This is particularly useful in situations where the anomaly being detected may be the result of a complex, multi-step process that unfolds over a long period of time.

### Long Short Term Memory

Long short-term memory (LSTM) networks are a type of recurrent neural network (RNN) that have proven to be particularly effective for tasks involving sequential data, such as language modeling, machine translation, and speech recognition, as well as anomaly detection. LSTMs are designed to overcome the limitations of traditional RNNs by introducing additional mechanisms for controlling the flow of information through the network [161].

In the context of anomaly detection, LSTMs can be used in a similar way to traditional RNNs to identify unusual patterns or events in data by learning a representation of normal behavior and then detecting deviations from this representation. This can be accomplished using techniques such as autoencoding or supervised or unsupervised classification, as described in the previous section [188].

Mathematically, LSTMs can be represented as follows:

$$
\begin{aligned}
\mathbf{f}_t &= \sigma(W_f \mathbf{x}t + U_f \mathbf{h}t - 1 + \mathbf{b}_f) \\
\mathbf{i}_t &= \sigma(W_i \mathbf{x}t + U_i \mathbf{h}t - 1 + \mathbf{b}_i) \\
\mathbf{o}_t &= \sigma(W_o \mathbf{x}t + U_o \mathbf{h}t - 1 + \mathbf{b}_o) \\
\mathbf{g}_t &= \tanh(W_g \mathbf{x}t + U_g \mathbf{h}t - 1 + \mathbf{b}_g) \\
\mathbf{c}_t &= \mathbf{f}t \odot \mathbf{c}t - 1 + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}
$$

where $\mathbf{x}_t$ is the input at time $t$, $\mathbf{h}_t$ is the hidden state at time $t$, $\mathbf{c}_t$ is the cell state at time $t$, $\mathbf{f}_t$, $\mathbf{i}_t$, $\mathbf{o}_t$, and $\mathbf{g}_t$ are the forget, input, output, and cell gates at time $t$, respectively, $W$ and $U$ are weight matrices, and $\mathbf{b}$ is a bias vector. The functions $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and hyperbolic tangent functions, respectively, and $\odot$ denotes element-wise multiplication [162].

The key difference between LSTMs and traditional RNNs is the introduction of the cell state $\mathbf{c}_t$ and the gates $\mathbf{f}_t$, $\mathbf{i}_t$, $\mathbf{o}_t$, and $\mathbf{g}_t$. The cell state is a long-term memory component that can store information over a longer period of time and is controlled by the forget and input gates. The forget gate determines what information from the previous cell state should be retained, while the input gate determines what new information should be added to the cell state. The output gate controls what information from the cell state should be output as the hidden state at the current time step. The cell state and gates allow LSTMs to selectively store and retrieve information over a long period of time, making them more effective at capturing long-term dependencies in data.

### 2.3.4 Generative Adversarial Networks

GANs typically predict anomalies by comparing the output of the discriminator for the new data to a predetermined threshold. If the output of the discriminator falls below the threshold, it is considered an anomaly. Mathematically, this can be expressed as follows:

$$
\text{Anomaly} = \begin{cases} 1 & \text{if } D(\mathbf{x}) < \text{threshold0} \\ \text{otherwise} \end{cases}
$$

where $D(\mathbf{x})$ is the output of the discriminator for the input data $\mathbf{x}$, and threshold is a predetermined threshold. The output of the discriminator can be interpreted as the probability that the input data is real.

The generator can learn to produce synthetic data that is indistinguishable from the genuine data, which is a problem when employing GANs for anomaly detection because the discriminator will always output high probability for both real and synthetic data. Researchers have suggested a number of solutions to this problem, including adding more constraints to the generator or utilizing various loss functions [57].

The GAN's ability to detect anomalies can be enhanced by incorporating the reconstruction loss during training to prevent the generator from developing the ability to produce artificial data that is indistinguishable from the real data.

Utilizing several loss functions is a potential strategy for enhancing GAN performance for anomaly detection. As an illustration, the Wasserstein GAN (WGAN) [62] is a GAN variant that employs the Wasserstein distance as the loss function and has been found

to enhance the stability and performance of GANs. According to this definition, the Wasserstein distance is:

$$\mathrm{W}(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} E_{(x,y)\sim\gamma}[||x - y||]$$

where $P$ and $Q$ are the real and synthetic data distributions, respectively, and $\Gamma(P, Q)$ is the set of all joint distributions with marginals $P$ and $Q$. The Wasserstein distance measures the minimum cost of transporting the real data distribution $P$ to the synthetic data distribution $Q$, where the cost is defined as the expectation of the distance between a real data sample and a synthetic data sample.

Using the Wasserstein distance as the loss function in a GAN has been shown to improve the ability of the GAN to detect anomalies.

In summary, GANs can be used for anomaly detection by training the generator on normal data and using the discriminator to detect anomalies in new, unseen data. Incorporating a reconstruction loss or using different loss functions, such as the Wasserstein distance, can improve the performance of GANs for this task.

### 2.3.5 Hybrid Models

#### LSTM Autoencoder

An LSTM Encoder-Decoder (LSTMED)[41] architecture consists of two main components: an encoder and a decoder.

The encoder processes the input time series data, $\mathbf{X}$, using an LSTM (Long Short-Term Memory) layer to capture long-term dependencies in the data. The output of the encoder is then passed to the decoder, which generates a reconstruction of the input data [133].

The LSTM Encoder-Decoder architecture is trained using the reconstruction error between the input data and the reconstructed data. The reconstruction error is used as a measure of the deviation from the normal behavior of the time series data [5].

After processing the input data through the LSTM layer, the encoder output is obtained as the final hidden state, $\mathbf{z} = \mathbf{h}_T$.

The decoder LSTM layer then processes the encoder output to reconstruct the input data:

$$\hat{\mathbf{h}}_t = \mathrm{LSTM}(\hat{\mathbf{h}}_{t-1}, \hat{\mathbf{x}}_{t-1})$$

where $\hat{\mathbf{h}}_t$ is the hidden state at time $t$ and $\hat{\mathbf{x}}_{t-1}$ is the input at time $t - 1$.

The decoder output is obtained using a sigmoid activation function:

$$\hat{\mathbf{x}}_t = \mathrm{sigmoid}(\mathbf{W}_o\hat{\mathbf{h}}_t + \mathbf{b}_o)$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ are the weights and biases of the output layer.

The reconstruction error is then calculated using a loss function, such as mean squared error (MSE) [3]:

$$\mathrm{MSE} = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t - \hat{\mathbf{x}}_t)^2$$

During training, the model parameters (e.g. weights and biases of the LSTM layers and the output layer) are optimized to minimize the reconstruction error. Once the model is trained, it can be used to detect anomalies in new time series data by comparing the reconstruction error with a threshold.

The distribution of the latent variables in the model is modeled using a Gaussian distribution in the LSTMED method. Using the "train Gaussian" set, the mean and covariance of the Gaussian distribution are learned from a portion of the training data. In order to rebuild the input sequences, the latent variables are sampled from the Gaussian distribution and fed through the LSTM model during training.

The Gaussian distribution is used in the model to capture intricate data dependencies and to create fresh samples by taking samples from the distribution. This is so that the structure of the data may be captured by the Gaussian distribution, which can represent a broad range of probabilities and has a well-defined mean and covariance.

Mathematically, let $z_i$ denote a latent variable sampled from the Gaussian distribution, with mean $\mu$ and covariance $\Sigma$. The probability density function of the Gaussian distribution is given by:

$$p(z_i) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(z_i - \mu)^T \Sigma^{-1} (z_i - \mu)\right)$$

where $k$ is the number of dimensions of the latent variable and $|\Sigma|$ is the determinant of the covariance matrix. Sampling from the Gaussian distribution allows the model to generate new latent variables that are likely to be similar to the latent variables in the training data. This is useful for generating new samples or for performing tasks such as anomaly detection, where the model needs to identify instances that are significantly different from the training data.

The latent variables are then passed through the LSTM model to reconstruct the input sequences, which can be written as:

$$\hat{x}_i = f(z_i)$$

where $\hat{x}_i$ is the reconstruction of the input sequence $x_i$ and $f(\cdot)$ is the LSTM model. The goal of training is to minimize the reconstruction error between the input and reconstructed sequences, which can be written as:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \hat{x}_i|^2$$

where $n$ is the number of input sequences and $|\cdot|$ denotes the Euclidean distance. By minimizing the reconstruction error, the model is able to learn a good representation of the input data, which allows it to generate new samples that are similar to the training data or to identify anomalous instances.

## 2.4 Explaining Anomalies

### 2.4.1 Causal Bayesian Network

A Bayesian network [180] is a probabilistic graphical model that represents the joint probability distribution of a set of random variables and their dependencies using a directed acyclic graph (DAG) [177]. In the context of cyber-physical systems, a Bayesian network [135] can be used to model the relationships between the various components of the system, such as sensors, controllers, actuators, and other devices, and to analyze the behavior of the system under different operating conditions.

Anomalies in cyber-physical systems can be detected using various techniques, such as statistical analysis, machine learning, and data mining. Once an anomaly is detected, it is important to explain its cause and to identify the factors that contributed to the anomaly. A Bayesian network can be used to provide a systematic and rigorous framework for explaining anomalies in cyber-physical systems [44].

Suppose we have a Bayesian network that models the behavior of a cyber-physical system with N random variables, denoted by $X_1, X_2, ..., X_N$. Let $P(X_1, X_2, ..., X_N)$ denote the joint probability distribution of these variables, and let $G = (V, E)$ denote the DAG that represents the conditional independence relationships between the variables.

Let $A$ denote an anomaly that has been detected in the system, and let $O_1, O_2, ..., O_M$ denote a set of observed variables that are related to the anomaly [95] We can use the Bayesian network to explain the anomaly by computing the posterior probability of the causes of the anomaly, given the observed variables:

$$P(X_C|O_1, O_2, ..., O_M, A) = \frac{P(O_1, O_2, ..., O_M, A|X_C)P(X_C)}{P(O_1, O_2, ..., O_M, A)} \tag{2.2}$$

where $X_C$ denotes the set of variables that may have caused the anomaly. The posterior probability $P(X_C|O_1, O_2, ..., O_M, A)$ represents the degree of belief that the variables in $X_C$ caused the anomaly, given the observed variables $O_1, O_2, ..., O_M$ and the fact that the anomaly occurred.

The likelihood term $P(O_1, O_2, ..., O_M, A|X_C)$ represents the probability of observing the anomaly and the observed variables, given the causes $X_C$. This term can be computed using the conditional probability tables (CPTs) associated with the nodes in the Bayesian network.

The prior term $P(X_C)$ represents the prior probability of the causes $X_C$, before taking into account the observed variables and the anomaly. This term can be specified by the domain expert based on their knowledge and experience.

The evidence term $P(O_1, O_2, ..., O_M, A)$ represents the probability of observing the anomaly and the observed variables, regardless of the causes. This term can be computed by summing over all possible values of the variables in the network:

$$P(O_1, O_2, ..., O_M, A) = \sum_{X_C} P(O_1, O_2, ..., O_M, A|X_C)P(X_C) \tag{2.3}$$

Once we have computed the posterior probability of the causes of the anomaly, we can use the Bayesian network to identify the factors that contributed to the anomaly. We can

do this by computing the posterior probability of each variable in the network, given the observed variables and the fact that the anomaly occurred:

$$P(X_i|O_1, O_2, ..., O_M, A) = \sum_{X_C \setminus X_i} P(X_C|O_1, O_2, ..., O_M, A) \tag{2.4}$$

This computation involves summing over all possible values of the variables in $X_C$, except for $X_i$, and marginalizing out the other variables in the network. The resulting posterior probability $P(X_i|O_1, O_2, ..., O_M, A)$ represents the degree of belief that variable $X_i$ contributed to the anomaly, given the observed variables $O_1, O_2, ..., O_M$ and the fact that the anomaly occurred.

The posterior probabilities of the variables can be used to rank the factors that contributed to the anomaly, and to identify the most likely causes of the anomaly. The domain expert can use this information to investigate the factors that contributed to the anomaly and to take corrective actions to prevent similar anomalies from occurring in the future.

### 2.4.2 Tree Based Methods

Gradient Boosting Machines (GBMs) are a class of machine learning models that combine multiple decision trees to make more accurate predictions. GBMs iteratively add decision trees to the model, and each new tree is trained to correct the errors of the previous trees. The final prediction is the sum of the predictions from all the decision trees.

GBMs are popular because of their high accuracy and ability to handle large datasets. Additionally, GBMs can be interpreted rigorously, which is important for explaining the model's predictions to humans.

The GBM model can be defined as:

$$F(x) = \sum_{m=1}^{M} \beta_m h_m(x) \tag{2.5}$$

where $x$ is the input data, $F(x)$ is the prediction function, $h_m(x)$ is the $m^th$ tree, $\beta_m$ is the weight given to the $m^th$ tree, and $M$ is the total number of trees in the model.

To fit the model to the training data, GBMs use gradient descent to minimise the loss function:

$$L(y, F(x)) = \sum_{i=1}^{n} l(y_i, F(x_i)) \tag{2.6}$$

where $y$ is the target variable, $l$ is the loss function, $n$ is the number of training samples, and $F(x_i)$ is the prediction of the model for the $i^th$ sample [125].

GBMs can be interpreted [91] by examining the importance of each feature in the model. Feature importance is calculated by measuring how much the model's accuracy decreases when a feature is randomly shuffled. The higher the decrease in accuracy, the more important the feature is to the model's prediction.

Feature importance can be calculated as:

$$\text{importance}(j) = \frac{1}{M} \sum_{m=1}^{M} \sum_{t=1}^{|T_m|} 1(v_{j,t} = j) \Delta L_m^t \tag{2.7}$$

where $j$ is the index of the feature, $M$ is the number of trees, $T_m$ is the set of terminal nodes for the $m^th$ tree, $v_{j,t}$ is the splitting variable for the $t^th$ node, and $\Delta L_m^t$ is the decrease in the loss function for the $t^th$ node of the $m^th$ tree.

In addition to feature importance, GBMs also provide information about the contribution of each feature to each decision tree in the model. This information can be used to understand how the model is making its predictions .

The importance of human-readable and actionable insights is that they allow workers in the CPS to quickly identify and correct anomalies. The rules provide a clear path for workers to follow, which reduces the time it takes to correct the anomaly and minimizes the impact of the anomaly on the system [166].

The rules are generated using a tree-based GBM, which provides a high level of accuracy and interpretability [141]. The GBM generates rules by evaluating the importance of each feature and the contribution of each feature to each decision tree in the model. The rules are generated by parsing the decision trees in the model and extracting the if-then rules.

The if-then rules generated by the GBM are expressed in terms of the input features and the predicted labels. The rules can be of the form "if feature 1 is greater than X and feature 2 is less than Y, then the predicted label is Z". The rules can be transformed into actionable insights by replacing the input features with specific values and the predicted label with a specific action.

## 2.5 Anomaly Localization

Lack of anomaly localization is one of the biggest problems with anomaly detection in cyber-physical systems. The ability to pinpoint the precise location or source of an anomaly within a system is known as anomaly localization. This ability is often essential for pinpointing the anomaly's underlying cause and taking the necessary corrective action.

Despite the significance of anomaly localization in cyber physical systems, the majority of the research done now just focuses on identifying anomalies rather than localizing them. The difficulties of effectively modeling the interactions between many variables and the complexity of the systems are typically to blame for this.

Due to its capacity to recognize intricate patterns in data, deep learning has recently become a vital tool for anomaly detection in cyber-physical systems. The lack of interpretability of the models, as stated in [22], is one of the major obstacles to employing deep learning for anomaly detection. Because it is challenging to comprehend how deep learning models make their predictions, they are frequently referred to as "black boxes." Due to this lack of interpretability, it may be challenging to identify the underlying cause of an anomaly and localize it to a particular area or source.

In cyber physical systems, the absence of anomaly localization can have detrimental effects because it might be challenging to locate the origin of an anomaly and implement a solution. Consider a chemical factory, for instance, where a sensor reading from a specific sensor shows an anomaly [28].

Without localization, it could be challenging to ascertain if the abnormality is brought on by a fault with the sensor or by another element, like a breakdown in another area of the plant. Delays in finding and fixing the issue may result from this, which could have

detrimental effects on the plant's efficiency and safety.

There have been several attempts to address this issue model agnostics explainers.

In one of these papers [145], an explainability method known as "Anchors" is presented. It offers high-precision, model-independent explanations of machine learning models. The model's predictions can be explained by using anchors, which can find a subset of features in a dataset that are sufficient to predict a specific outcome. In addition to being exceedingly precise, anchors are model-agnostic, which means they can be used to determine the minimal subset of features that adequately explain a machine learning model's prediction.

The scope limitation of Anchors, which may prevent them from explaining broad patterns in a model's behavior, is just one of several drawbacks. They can only explain specific predictions. Additionally, Anchors don't always offer justifications for why a particular forecast was made and need a certain amount of feature engineering to be effective.

High precision and the ability to explain a model's forecast in terms that anyone can comprehend are two advantages of anchors.

Another technique used in machine learning to describe the output of black-box models is called SHapley Additive exPlanations (SHAP), which is cited in [112]. It is based on the game theory idea of Shapley value, which gives each feature a score to indicate how much that feature contributes to the model's output. As a result, it becomes an effective tool for comprehending and evaluating the choices made by machine learning models.

The advantages of SHAP include its ability to handle interactions between features, its ability to explain the model's behavior thoroughly, and its capability to explain the model's output in terms of specific characteristics. As a result, it serves as a convenient tool for comprehending the choices made by sophisticated models, including those employed in predictive analytics and computer vision.

The Shapley value's underlying assumptions, that may not always hold true, is one of SHAP's weaknesses. Additionally, it makes significant assumptions about the data, which could limit its applicability to data from the actual world. Accurate explanations also need a lot of computer power to develop, which presents a substantial challenge for real-world applications.

## 2.6 Conclusion

Finding anomalies in cyber-physical systems is essential for guaranteeing their dependable and secure operation. High-dimensional data and complicated interdependencies across data series cannot be handled by conventional approaches for anomaly identification, such as one-class SVMs and kernel density estimation. Deep learning-based algorithms for cyber-physical system anomaly detection have been developed recently, and these methods have shown excellent performance in a number of applications.

Deep learning-based approaches have the advantage of being able to automatically extract features from raw data, eliminating the need for manual feature engineering. These techniques are also good at capturing complicated patterns in the data, which makes them suitable for finding anomalies in high-dimensional and multivariate data.

A variety of deep learning models, including autoencoders, variational autoencoders, generative adversarial networks, and recurrent neural networks, have been used to detect anomalies in cyber-physical systems. These models have been used to find anomalies in

time series data, sensor data, and picture data among other data types.

There has been study on the use of hybrid approaches, which integrate deep learning with different methods like clustering or rule-based systems, in addition to deep learning models for anomaly identification. These methods may help anomaly detection systems perform better and be more reliable.

Overall, deep learning has demonstrated excellent results when used for anomaly detection in cyber physical systems, and it has the potential to dramatically raise the dependability and safety of these systems.

The difficulties of anomaly detection in the actual world, particularly the existing literature's neglect of the justification for observed abnormalities, require further study. A more thorough pipeline is required that not only finds anomalies but also confirms their existence and offers a justification for why they occur. This kind of paradigm is essential for bridging the gap between academic inquiry and real-world application.

# Chapter 3

# Related Work

The researchers employed RNNs, more specifically LSTM networks, to estimate future values in the data and recognize variations from the expected values as anomalies in [53]. In order to provide a broad dataset for testing, the researchers developed the TEP model in Python and used it to simulate numerous cyberattacks.

The researchers employed the Numenta Anomaly Benchmark (NAB) metric [98], which rates the accuracy and speed of anomaly identification, to assess the effectiveness of their approach. They compared the outcomes of their RNN-based approach to a conventional one that used dynamic principal component analysis (DPCA), a popular technique for defect detection in the TEP process [96]. In terms of early anomaly identification, they discovered that the RNN-based strategy performed better than the DPCA approach, and they offered a comparison of the outcomes using the NAB measure.

Overall, the research demonstrated the potential for using RNNs, specifically LSTM networks, for detecting anomalies in industrial multivariate time series data, and highlighted the importance of early detection in the context of cyber attacks on industrial control systems.

This [30] research proposes a novel approach for anomaly detection in industrial processes using a deep autoencoder [196] architecture based on 1D convolutional neural networks (CNNs) [167]. The approach relies on unlabeled data and involves splitting the autoencoder latent space into discriminative and reconstructive latent features [78]. An auxiliary loss based on k-means clustering is introduced for the discriminatory latent variables, and a top-K clustering objective is used to select the most discriminative features from the latent space. The approach is shown to improve downstream tasks such as anomaly detection, binary classification, and multi-class classification compared to standard autoencoder architectures.

The approach is intended to be used in the context of intelligent condition monitoring of industrial processes, where data-driven methods are needed to analyze changes in process parameters and detect anomalies that could impact the reliability of the system. Unsupervised or semi-supervised learning methods, such as those based on autoencoder frameworks or generative adversarial networks, are particularly useful in this context because they do not require labeled data, which can be difficult to obtain in industrial settings due to various constraints.

A method for anomaly detection using Generative Adversarial Networks (GANs)[37] is presented by the researchers in the study cited as [186]. The method uses a Bidirectional Generative Adversarial Network (BiGAN) [45] model, which may be utilized for unsupervised learning and can develop a rich feature representation for different data distributions. A generator network (G) and a discriminator network (D) that are trained to compete against one another in a minimax game make up the BiGAN model. The discriminator network is trained to distinguish between real and fake samples, while the generator network is trained to produce examples that are comparable to the training data.

The researchers suggest two scoring functions, G-score and D-score, to calculate the anomaly of new data samples when using the BiGAN model for anomaly detection. The G-score is based on the generator network's reconstruction error, whereas the D-score is based on the discriminator network's output. A sample is deemed anomalous if its G-score or D-score is higher than a specific critical value.

The findings demonstrate that the proposed BiGAN-based technique is much faster and has competitive performance when compared to other GAN-based approaches.

The paper [77] proposes a method for detecting abnormal signals in industrial multi-sensor signals, which are collected from multiple sensors arranged in a specific configuration. These signals typically consist of three types: regular signals that reflect normal conditions, defect signals that indicate faults or damage, and abnormal signals that are interference signals under normal operating conditions. Abnormal signals can interfere with the detection of defect signals and cause problems in industrial measurement and detection.

To address this problem, the proposed method involves several steps. First, the signals are preprocessed to stabilize them. Then, a stack spatial-temporal autoencoder [184] is used to extract features and reconstruct the signals. This autoencoder is based on improved deep stack long short-term memory and autoencoder feature extractors. Next, a high-dimensional unsupervised clusterer [124] is applied to detect abnormal signals.

The DeepAnT method [123] is a deep learning approach for detecting anomalies in time series data. It is designed to be able to detect a wide range of anomalies, including point anomalies [194], contextual anomalies [68], and discords.

The method consists of two modules: a time series predictor and an anomaly detector. The time series predictor module uses a deep convolutional neural network [193] to predict the next time stamp based on a window of time series data, referred to as the context. The predicted value is then passed to the anomaly detector module, which determines whether the time stamp is normal or abnormal.

DeepAnT is able to be trained even when anomalies are present in the data, and it can handle relatively small data sets due to the effective parameter sharing of the CNN. In experimental evaluations, it has been shown to outperform state-of-the-art anomaly detection methods on most of the benchmarks tested, while performing on par with others. It is a useful approach for detecting anomalies in time series data, particularly in the context of streaming data and the Internet of Things (IoT).

Two options are offered in [84]. Leveraging sparsely connected recurrent neural networks

(S-RNNs), multiple autoencoders are constructed using an independent framework as the first solution, as noted in [39]. These autoencoders have various neural network connection architectures, and by combining their predictions, the quality of outlier detection as a whole is increased. By employing an ensemble of autoencoders, the effects of overfitting to outliers, which can happen when using a single autoencoder, are minimized [33].

A common framework, which is the second option, also entails creating numerous autoencoders utilizing S-RNNs. The autoencoders' hidden representations, as opposed to their predictions, are integrated in this instance. By utilizing the complementing qualities of the several autoencoders, this system attempts to increase the robustness of the outlier detection.

The authors run experiments on two real-world time series data sets, including univariate and multivariate time series, to see how well the suggested solutions work. These studies' findings indicate that the suggested frameworks are capable of outperforming both standard methods and cutting-edge techniques for outlier detection in time series data.

The core idea behind OmniAnomaly [164] is to learn robust representations of the normal patterns in the data using techniques such as stochastic variable connection and planar normalizing flow. These techniques help to capture the stochasticity in the data more accurately than deterministic variables. Once the normal patterns have been learned, the input data is reconstructed from these representations, and the reconstruction probabilities are used to identify anomalies. This approach allows OmniAnomaly to detect anomalies at the entity level, rather than at the metric level, which can be more intuitive, effective, and efficient than analyzing each metric separately.

The effectiveness of the OmniAnomaly method has been evaluated on three real-world datasets, including two public datasets from the aerospace industry and a new dataset from an internet company that consists of server machine data. The results of these experiments show that OmniAnomaly achieves an overall F1-score of 0.86, significantly outperforming the best-performing baseline method.

NASA researchers suggest using LSTM to identify anomalies in spacecraft telemetry data [72]. In order to decrease the frequency of false positives, the authors also suggest a complementary unsupervised and nonparametric anomaly thresholding strategy. False positives or false alarms are a concern for anomaly detection systems since they increase the burden for operations engineers and lower the system's overall reliability [198]. The proposed thresholding method is predicated on the idea that while the distribution of reconstruction errors in normal data tends to follow a Gaussian distribution, it is possible for this distribution to deviate in the case of anomalous data. The method can successfully reduce false positives by choosing a threshold based on the Gaussian distribution.

On the basis of telemetry data from the Mars Science Laboratory (MSL) rover, Curiosity, and the Soil Moisture Active Passive (SMAP) satellite [50] and [60] respectively, the proposed LSTM-based and thresholding approaches are assessed. The evaluation's findings demonstrate that the LSTM-based methodology can detect anomalies with a high degree of accuracy and is easier to understand than other approaches. Reduced false positives are another benefit of the unsupervised thresholding method. The authors also provide methods for further reducing false positives, such as using an expert panel to examine the findings and giving more context information.

All things considered, the suggested techniques show the potential of LSTM networks for anomaly identification in satellite telemetry data. Combining the thresholding approach with the LSTM-based approach enables the detection of abnormalities in a scalable and understandable manner while also significantly lowering false positives. These techniques may lessen the workload for operations engineers and enhance spacecraft monitoring.

TadGAN [57] is a method for detecting anomalies in time series data using a combination of LSTM recurrent neural networks and GANs. In this approach, LSTM networks [158] are used as the base models for the generator and critic in the GAN. The GAN [38] is trained using cycle consistency loss, which allows for the effective reconstruction of time series data. The authors also propose a number of methods for computing reconstruction errors, as well as different approaches for combining these errors with critic outputs to compute anomaly scores.

To evaluate the performance of TadGAN, the authors compare it to eight baseline anomaly detection methods on 11 datasets from a variety of sources, including NASA, Yahoo, Numenta, Amazon, and Twitter. The results show that TadGAN is effective at detecting anomalies and outperforms the baseline methods in most cases, with the highest averaged F1 score across all datasets.

In addition to demonstrating the effectiveness of the TadGAN approach, the authors also provide several contributions to the field of time series anomaly detection. These contributions include the use of LSTM networks for time series reconstruction, the incorporation of cycle consistency loss into the training process, and the proposal of a number of novel methods for computing reconstruction errors and anomaly scores.

GLAD [75] is a human-in-the-loop learning algorithm that enables the use of simple, explainable global anomaly detectors in practice. It does this by adapting the global detectors to be more relevant to specific data instances in different parts of the input feature space. This is accomplished through label feedback from human analysts, who label instances as either normal or anomalous.

The algorithm works by first training a neural network on unlabeled instances to place a uniform prior on the relevance of each member of the anomaly detection ensemble over the input feature space. The neural network is then fine-tuned using labeled instances to adjust the local relevance of each ensemble member.

One key aspect of GLAD is that it provides explanations to the human analysts, which can improve their understanding of the anomalies being detected. This is accomplished through the use of a relevance neural network [144], which learns the local relevance of each ensemble member and provides a visual explanation of how the ensemble members contribute to the overall anomaly score for a given instance.

# Chapter 4

# Tennessee Eastman Process: Key Terminology and Context

## 4.1 The Tennessee Eastman Process

The Tennessee Eastman process (referred to in this paper as TEP) has been widely used as a testbed to study various challenges faced in continuous processes. Originally proposed by Downs and Vogel (1993) [48], the TEP has been used for plant-wide control design, multivariate control, optimisation, predictive control, adaptive control, nonlinear control, process diagnostics, and educational purposes. Several control designs have been proposed for the TEP, including those by Ricker (1996) [146] and Bathelt (2015) [14]. In recent years, many studies involving the TEP have focused on fault detection using classical statistics or machine learning methods.

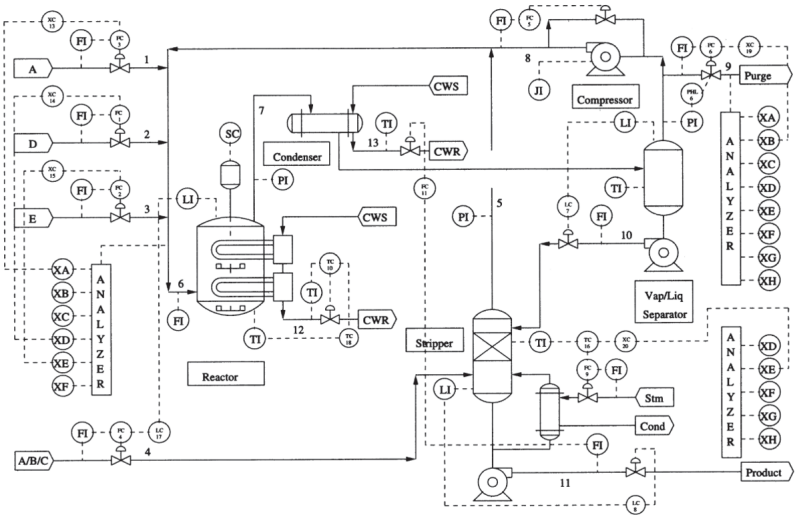



Figure 4.1: Tennessee Eastman Process [64]

There are two main types of variables in the Tennessee Eastman process: measured variables (XMEAS) and manipulated variables (XMV). Measured variables are the pro-

cess outputs that are monitored and recorded by the system. These variables provide information about the state of the process and can be used to detect abnormal behavior or faults. There are 41 measured variables A.3 in the Tennessee Eastman process.

Manipulated variables, on the other hand, are the inputs that can be adjusted by the system to control the process. These variables can be changed to correct any faults or to optimize the process. There are 12 manipulated variables 4.3 in the Tennessee Eastman process.

## 4.2 What are Features in Tennessee Eastman?

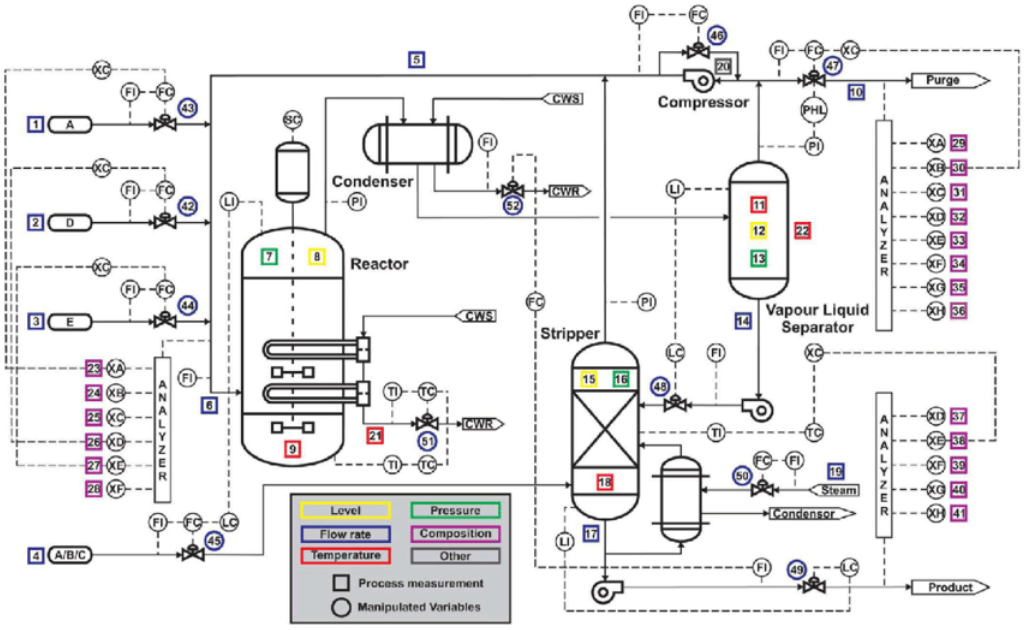



Figure 4.2: Tennessee Eastman Feature Key [32]

In the context of the Tennessee Eastman Process (TEP) cyber physical system, a feature is a specific variable or metric that is measured or calculated from the system's sensor data. These features are derived from raw sensor measurements and can represent different aspects of the system's operation, such as temperature, pressure, or flow rate. For example, a feature could be the temperature of a particular reactor vessel or the flow rate of a specific gas stream. The feature extraction process involves pre-processing the raw sensor data to extract and transform these features into a suitable format for the anomaly detection model.

A sensor measurement is a specific value obtained from a sensor, such as a temperature or pressure reading. While a sensor measurement can be considered a feature, not all features are directly derived from sensor measurements. For example, a feature may be a calculated metric based on multiple sensor measurements or other data sources. In the TEP system, sensor measurements are used to calculate the various features that represent different aspects of the system's operation.

In the TEP cyber physical system, features are related to components in the system because they represent specific aspects of the system's operation. For example, a feature may be related to a specific sensor or actuator, or it may represent the behavior of a particular subsystem within the system. This relationship between features and system components is important because it can help to identify the root cause of an anomaly or fault.

### 4.2.1 Tennessee Eastman Feature Key

#### Level

Level refers to the measurement of the height of liquid or gas in a tank or vessel. In the Tennessee Eastman process, the level feature may refer to the level of the reactor, the feed tank, or other vessels in the process.

#### Flow rate

Flow rate refers to the amount of fluid that passes through a given point per unit time. In the Tennessee Eastman process, the flow rate feature may refer to the flow rate of reactants or products in various parts of the process, such as the feed flow rate, cooling water flow rate, or the flow rate of gas or liquid in the reactor. Temperature: Temperature refers to the measure of the average kinetic energy of the molecules in a substance. In the Tennessee Eastman process, the temperature feature may refer to the temperature of the reactor, the cooling water temperature, or the temperature of the feed stream.

#### Pressure

Pressure refers to the force exerted by a fluid on the walls of the container that it is in. In the Tennessee Eastman process, the pressure feature may refer to the pressure in the reactor, the feed tank, or other vessels in the process. Composition: Composition refers to the proportion of different chemical components in a substance or mixture. In the Tennessee Eastman process, the composition feature may refer to the composition of the feed stream, the product stream, or the reactor contents.

#### Other

The "Other" category refers to other relevant features or variables in the Tennessee Eastman process that do not fall under the categories of Level, Flow rate, Temperature, Pressure, or Composition. Examples of features that may fall under the "Other" category could be the pH level, the stirring rate, or the viscosity of the reactor contents, among others.

#### Process measurements

Process measurements refer to the various sensors and instruments used to measure and monitor the different process variables in the Tennessee Eastman process. These measurements provide feedback about the current state of the process and are used to control the process variables. Examples of process measurements in the Tennessee Eastman process

include the sensors that measure the temperature, pressure, flow rate, level, and composition of various streams in the process.

### Manipulated variables

Manipulated variables refer to the variables that are actively controlled by the process control system to achieve the desired process objectives. These variables can be adjusted up or down to achieve a particular target value or setpoint. Examples of manipulated variables in the Tennessee Eastman process include the feed flow rate, cooling water flow rate, heating or cooling duty, and reactor stirring rate, among others. By manipulating these variables, the control system can achieve the desired process objectives, such as maintaining a certain product quality, maximizing the reactor efficiency, or minimizing the production of unwanted byproducts.

| **Variable name** | Number | Base value | Units |
|---|---|---|---|
| D feed flow | XMV(1) | 63.053 | $kg\,h^{-1}$ |
| E feed flow | XMV(2) | 53.980 | $kg\,h^{-1}$ |
| A feed flow | XMV(3) | 24.644 | ks cm h |
| A and C feed flow | XMV(4) | 61.302 | ks cm h |
| Compressor recycle valve | XMV(5) | 22.210 | % |
| Purge valve | XMV(6) | 40.064 | % |
| Separator pot liquid flow | XMV(7) | 38.100 | $m^3\,h^{-1}$ |
| Stripper liquid product flow | XMV(8) | 46.534 | $m^3\,h^{-1}$ |
| Stripper steam valve | XMV(9) | 47.446 | % |
| Reactor cooling water flow | XMV(10) | 41.106 | $m^3\,h^{-1}$ |
| Condenser cooling water flow | XMV(11) | 18.114 | $m^3\,h^{-1}$ |
| Agitator speed | XMV(12) | 50.000 | rpm |

Figure 4.3: Tennessee Eastman Process Manipulated Variables[64]

## 4.3 Anomalies in Tennessee Eastman

An anomaly in the TEP cyber physical system can correspond to a fault, but not all anomalies necessarily indicate a fault. An anomaly is any deviation from the expected or normal behavior of the system, which may be caused by a fault, an attack, or other unexpected events. In the context of the TEP system, faults may include equipment malfunctions, sensor errors, or control system failures. Different types of faults may correspond to different subsets of features, and the anomaly detection model should be designed to detect and localize these different types of faults.

| Fault | Fault Type | Description |
|---|---|---|
| Fault A | Step | A/C Feed Ratio, B Composition Constant |
| Fault B | Step | B Composition, A/C Ratio Constant |
| Fault C | Step | Reactor Cooling Water Inlet Temperature |
| Fault D | Step | Condenser Cooling Water Inlet Temperature |
| Fault E | Step | A Feed Loss |
| Fault F | Random Variation | A, B, C Feed Composition |
| Fault G | Random Variation | C Feed Temperature |
| Fault H | Random Variation | Reactor Cooling Water Inlet Temperature |
| Fault I | Slow Drift | Reaction Kinetics |
| Fault J | Sticking | Reactor Cooling Water Valve |
| Fault K | Unknown | Unknown |
| Fault L | Unknown | Unknown |
| Fault M | Unknown | Unknown |
| Fault N | Unknown | Unknown |

Table 4.1: Table of Faults in Simulated Data

### 4.3.1 Types of Faults

The example faults shown in Fig 4.4 serve as a practical illustration of the various types of faults shown in Table **??**. These faults are used as a means of demonstrating and clarifying the different categories of faults that can occur in the Tennessee Eastman process. By showcasing these specific faults, we hope the reader can better understand the characteristics and implications of each fault type, and how they can affect the overall system performance.

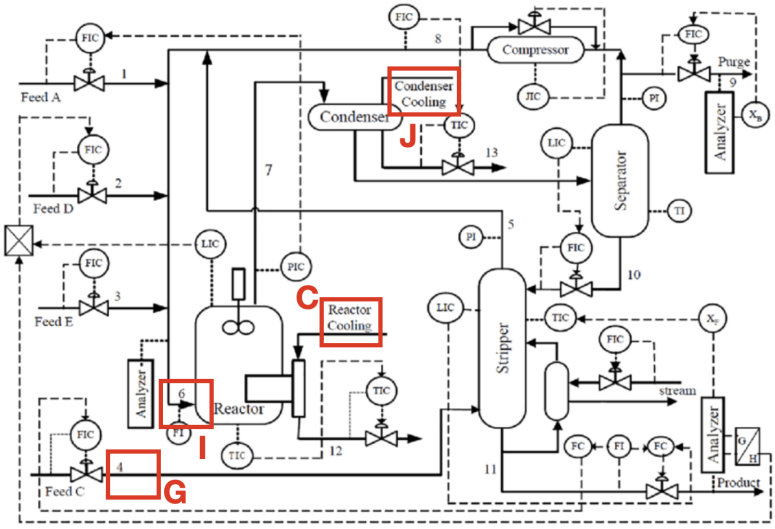



Figure 4.4: Tennessee Eastman Process Fault Examples

The fault plots displayed in this section were obtained by simulating a single type of fault repeatedly over a defined time period. The selected examples were specifically chosen for their conspicuousness, making them ideal for illustrative purposes. By using

these examples, the faults are readily observable, enabling a clear demonstration of the implications and characteristics of each fault type.

### Step Faults

Step faults can arise due to sudden changes in valve position, which alters the flow rate or pressure of a stream. An example of this is a valve closing abruptly in the reactor feed line, leading to a sudden drop in reactant flow rate.

### Example

Here, we demonstrate an example of a step fault, labeled as C in Figure 4.4 and listed as Fault C in Table 4.1. The fault occurs in the reactor subsystem, where the temperature of the cooling water inlet is abruptly increased to produce a step fault in the system. This fault is detected by the sensors that monitor both the temperature in the reactor and the flow of cooling water. The impact of the step fault can be observed in Figure 4.5.

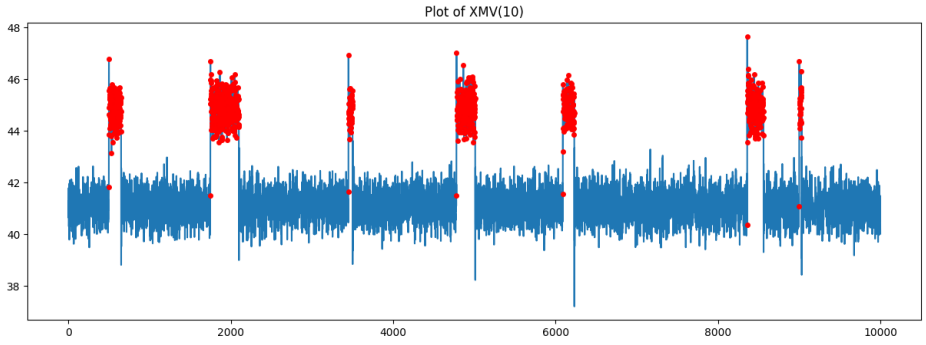



Figure 4.5: Step Fault Example

### Random Variation Faults

Random variation faults can occur due to sensor measurement errors, leading to fluctuations in the process variables over time. An example of this is noise interference in the temperature measurement of a reactor, leading to erroneous temperature readings.

### Example

The fault occurs in the Feed C stream, which is labeled as G in Figure 4.4 and listed as Fault G in Table 4.1. The temperature of the Feed C stream fluctuates randomly, and it flows into the Stripper subsystem. Consequently, the temperature sensor in the Stripper system records random variations in temperature, corresponding to the changes in the temperature of the Feed C stream. These random variations in temperature are clearly visible in Figure 4.6.

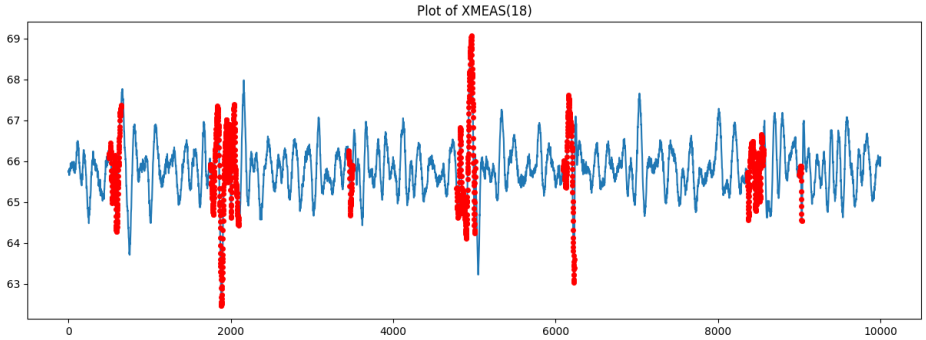



Figure 4.6: Random Variation Faults

### Slow Drift Faults

Slow drift faults can occur gradually over time, such as catalyst deactivation or fouling in the reactor, resulting in a decrease in conversion or selectivity of the reaction. An example of this is catalyst deactivation causing a gradual decline in the reactor's conversion efficiency.

### Example

In the case of Fault I, which is labeled as I in Figure 4.4 and listed as Fault I in Table 4.1, we focus on the reaction kinetics in the reactor system. Reaction kinetics in this context refers to the study of the rate and mechanism of chemical reactions that occur in the reactor. Thus, the sensor that monitors the kinetics in the system is alerted when the rate of the reaction deviates from the expected value. The deviation can be observed in Figure 4.7, which clearly shows a slow drift in the reaction rate as the fault progresses.

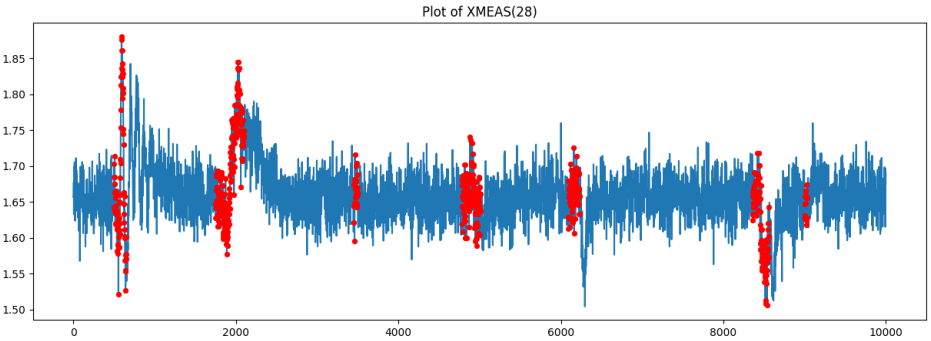



Figure 4.7: Tennessee Eastman Process Fault Examples

### Sticking Faults

Sticking faults can occur due to a valve getting stuck in a partially open or closed position, causing a fixed duration and magnitude change in the process variables. An example of this is a control valve getting stuck in a partially open position, leading to a drop in reactor pressure.

### Example

In this section, we examine the sticking fault, which is labeled as J in Figure 4.4. In this fault scenario, the condenser cooling water valve becomes stuck in a fixed position, resulting in rapid and unpredictable changes in the pressure of water flowing into the Condenser subsystem. The effects of this fault are clearly visible in Figure 4.8.

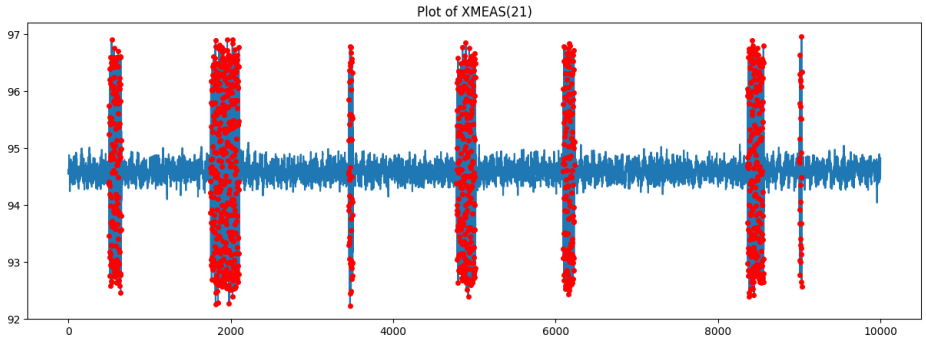



Figure 4.8: Tennessee Eastman Process Fault Examples

### Unknown faults

Unknown faults can occur due to a malfunctioning actuator or controller, affecting the set-point or feedback of a loop. An example of this is a malfunctioning control valve actuator, leading to erratic changes in the process variables with no apparent cause.

## 4.4 Localising Anomalies

The process of localising a fault in the TEP system involves identifying the specific component or subset of components that are responsible for the anomalous behavior detected by the model. This can help to identify the root cause of the fault and enable targeted corrective actions to be taken. For example, if a fault is detected in a particular reactor vessel, localising the fault to this component can help to identify potential equipment malfunctions or other issues that may be causing the anomalous behavior.

As an illustration, let us consider the step fault scenario described in section 4.3.1. In this case, the sensor monitoring the temperature in the cooling water inlet to the reactor detected the anomaly. Since this inlet is not connected to any other system, we can confidently infer that the fault occurred within the reactor subsystem, specifically within the inlet flow itself.

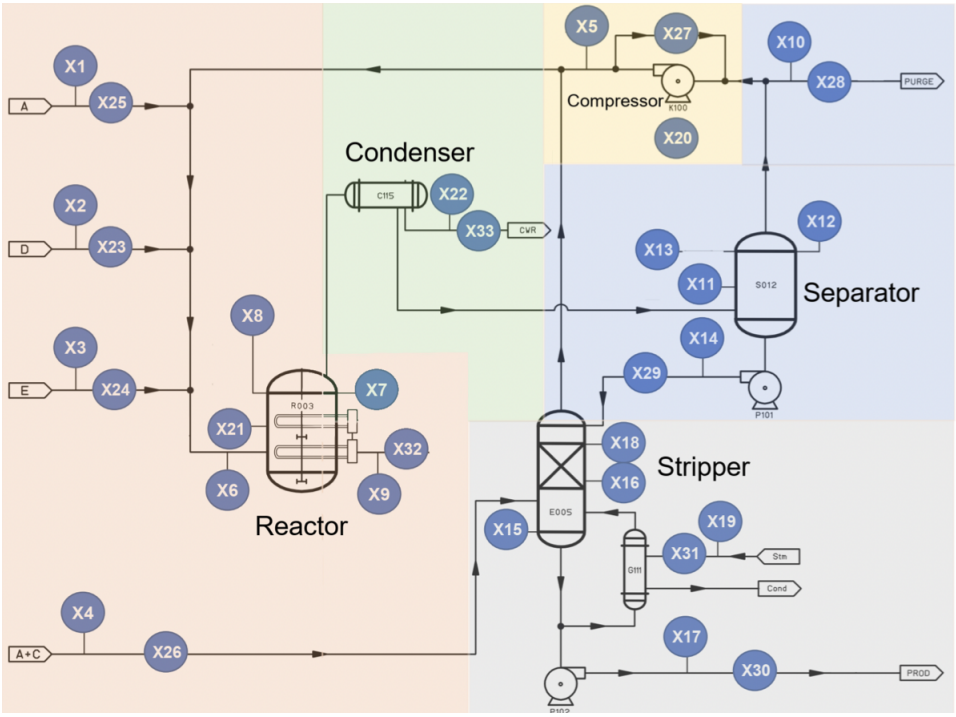



Figure 4.9: Tennessee Eastman Process Subsystems [119]

## 4.5 Explaining Anomalies

Fault explanation is the process of identifying the root cause of a fault in a system, including any contributing factors or environmental conditions, and providing a clear and understandable description of the problem to relevant parties. This involves identifying the specific component or subsystem responsible for the fault, as well as any relevant measurements or observations that may aid in the explanation process. The goal of fault explanation is to enable stakeholders to understand the cause of the fault and make informed decisions regarding remediation and prevention strategies.

Let us take the Random Variation fault discussed earlier as another example. This fault originated in the Feed C system, but it was detected and localized in the inflow to the Stripper subsystem. While this information may not be effective in isolation, it becomes more meaningful when we consider the causal relationship between the Stripper and the Feed systems.

The Feed C system is connected to the Stripper system, and changes in the Feed C temperature can affect the temperature in the Stripper. Thus, when the sensor monitoring the temperature in the Stripper system detected random variations, it was reasonable to infer that the fault was in the Feed C system. By localizing the fault to the Feed C system, operators can investigate potential causes such as fluctuations in the temperature of the Feed C stream or issues with the sensors or actuators in that system.

By providing a clear explanation of the fault, stakeholders can better understand the problem and take appropriate actions to address it.

## 4.6 Conclusion

In conclusion, this chapter has provided a comprehensive introduction to the Tennessee Eastman process and its role as a testbed for anomaly detection techniques. We have covered the various subsystems that make up the TEP, and illustrated the complexity of the process through examples of different types of faults that can occur.

Furthermore, we have emphasized the importance of localising and explaining faults within the TEP, as this is a critical step in the anomaly detection process. This chapter has also highlighted the significance of the TEP as a realistic and challenging environment in which to evaluate the effectiveness of anomaly detection methods.

The purpose of this chapter was to provide the reader with the necessary context and information to better understand the methodology and reasoning employed in the rest of this thesis.

# Chapter 5

# Proposed Methodology

In this section, it is important to note that all examples presented will be conducted by adjusting the Reactor Cooling Water Flow in the system (feature XVM(10)), unless explicitly stated otherwise. This fault is labeled as C in Figure 4.4 and listed as Fault C in Table 4.1 for reference. One of the main reasons for choosing this parameter is that it has been identified as a critical variable that causes a change point anomaly in the system. This characteristic makes it useful for visualization and demonstration purposes, as it allows us to observe the effects of different operational conditions on the system's output in a more pronounced manner. The effect of this example can be observed in Figure 4.5

## 5.1 Data Generation

The Tennessee Eastman process is a complex chemical plant composed of several interconnected units, each having unique sets of operating parameters and process variables. In order to analyse the behavior of this process under different operating conditions, it is crucial to obtain significant amounts of data that accurately replicate the real-world process.

To address this need, a simulation of the Tennessee Eastman process has been developed using Fortran code [182] that was compiled into Python code [67].

To increase the versatility of the simulation, a Python class was created on top of the Fortran code. The class permits users to configure the simulation according to their desired data specifications. The user can select from a list of faults, such as step faults, random variation faults, slow drift faults, sticking faults, and unknown faults, known to occur in the Tennessee Eastman process.

A significant characteristic of our simulation is its ability to determine the intervals at which faults occur. This feature enables the generation of both short and long faults, spaced out at different intervals, including simultaneous overlapping faults. Varying fault intervals, users can create data that represents a wide range of scenarios and operating conditions.

In addition to generating fault-included data, the simulation also generates training data that does not contain faults. This training data is simulated for a predetermined period, allowing the model to understand the regular operating behavior of the process.

To evaluate the model's effectiveness, labels for the data are created using the config-

ured fault intervals. These labels are not used during training but serve as a means of testing. By comparing the predicted faults with the actual faults, we can determine the model's accuracy and effectiveness in identifying, locating, and explaining faults in the Tennessee Eastman process.

Simulating the data, instead of using publicly available data, is beneficial as it provides a ground truth of the data, which is crucial for evaluating the system's performance. Additionally, simulating the data also allows for complete control over the fault scenarios and generating more detailed and realistic scenarios. This creates a dataset that can truly test the limits of the system and push it to its full potential. Furthermore, simulating the data also enables having an unlimited amount of data, which is crucial for training and testing large deep learning networks.

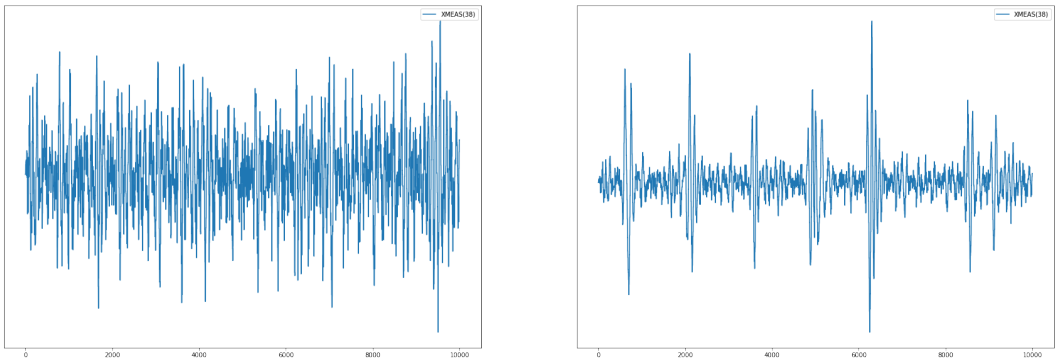



Figure 5.1: Normal vs Anomalous Data in sensor XMEAS(38) for Fault F

In summary, simulating the data instead of using publicly available data allows for complete control over the fault scenarios and generating more detailed and realistic scenarios. Furthermore, having the ground truth of the data and the ability to generate an unlimited amount of data is crucial for training and testing large machine learning models.

## 5.2 Data Preprocessing

In the initial stage of the pipeline, the data is processed and prepared for use in a neural network. Certain columns may be removed from the data in order to eliminate unnecessary or irrelevant information.

It is important to then assess the presence of missing values in the data and employ imputation techniques as needed [147][47]. The imputation of missing values is crucial as many machine learning models are not robust to the presence of missing data. However, the user has the option to specify an alternative imputation method.

Finally, the data may be scaled in order to balance the influence of features with large scales. The scaling equation is as follows:

$$x' = \frac{x - \mu}{\sigma} \tag{5.1}$$

where $x$ is the original value, $x'$ is the scaled value, $\mu$ is the mean of the data, and $\sigma$ is the standard deviation of the data [157].

By implementing these preprocessing steps, the data is suitably transformed for use in an anomaly detection algorithm, ultimately leading to improved accuracy and reliability of the resulting models.

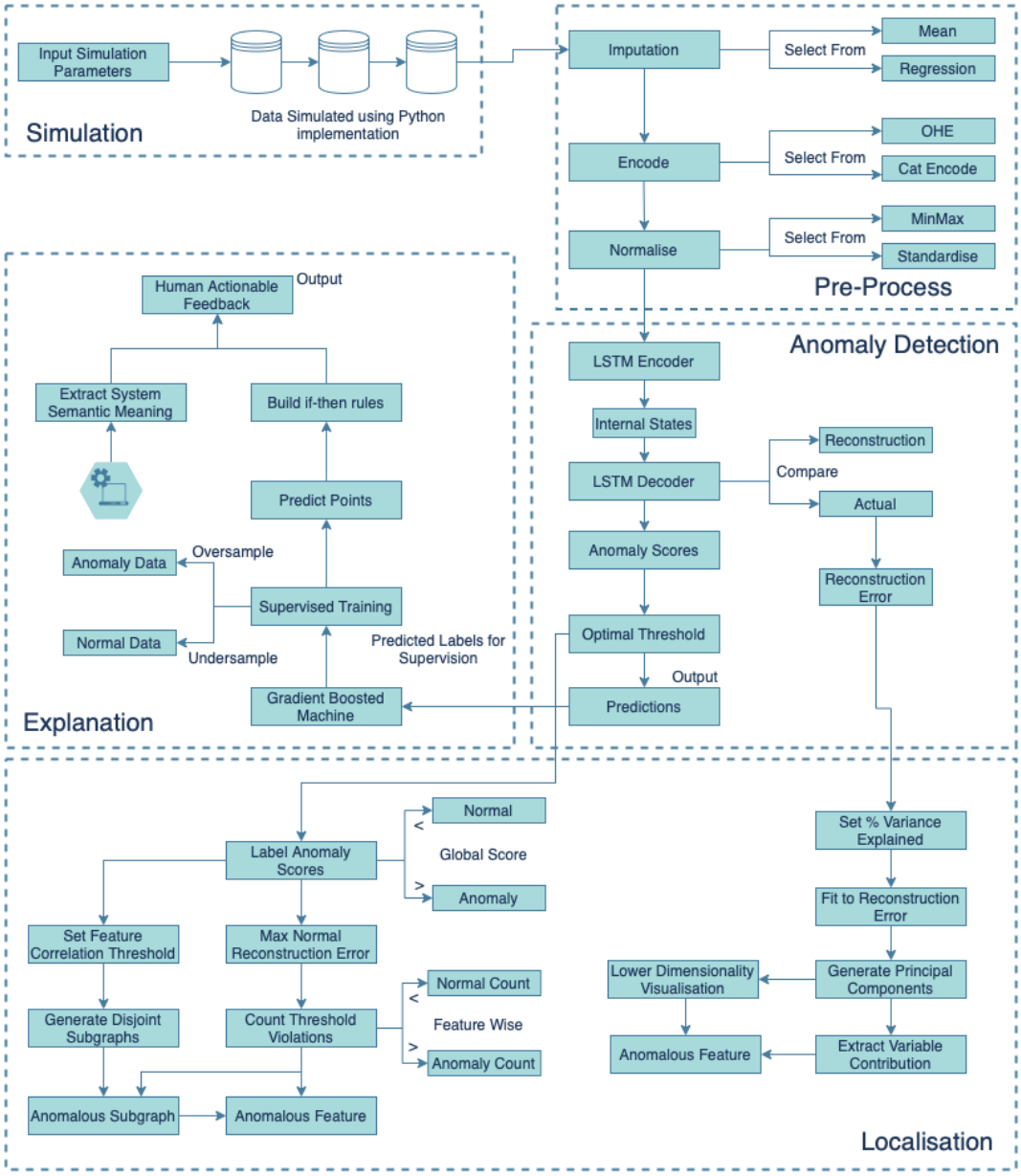



Figure 5.2: CADLAE Architecture

## 5.3 Seq2Seq LSTM Encoder Decoder

An LSTM Encoder-Decoder (LSTMED)[41] architecture consists of two main components: an encoder and a decoder. The encoder-decoder approach is a methodology used

in a variety of models such as autoencoders, recurrent neural networks, long short-term memory, and generative adversarial networks. It involves encoding input data into a lower-dimensional representation and decoding it to reconstruct the original input or generate new instances. Autoencoders use it for unsupervised learning and reconstruction, while RNNs and LSTMs use it for sequence-to-sequence learning. GANs use it to generate new instances that follow a generative distribution. For a more indepth analysis of these approaches please refer to Section 2.3

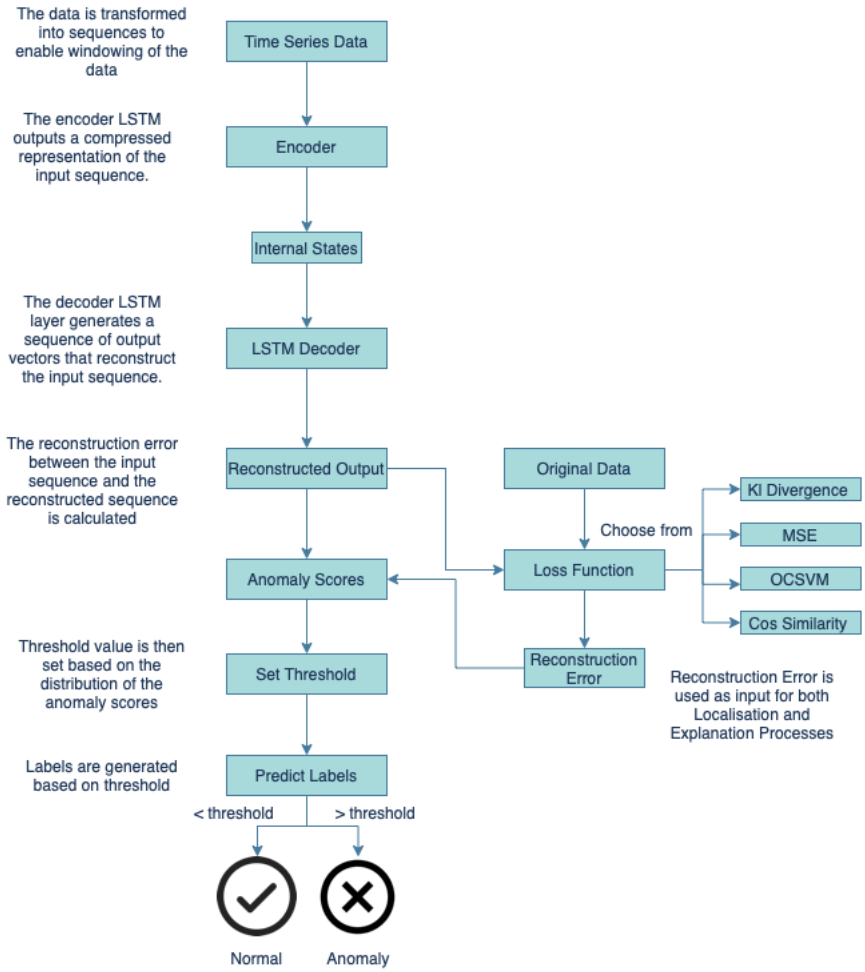



Figure 5.3: CADLAE Architecture: LSTMED

The LSTM Encoder-Decoder architecture is trained using the reconstruction error between the input data and the reconstructed data. The reconstruction error is used as a measure of the deviation from the normal behavior of the time series data [5]. The equations for the LSTMED architecture have been defined previously in 2.3.3.

## 5.4 Optimizing Threshold

The Receiver Operating Characteristic (ROC) [51] curve is a widely used metric for evaluating binary classification problems. It is a graphical representation of the True Positive Rate (TPR) versus the False Positive Rate (FPR) at different classification thresholds. The ROC curve provides insight into the trade-off between TPR and FPR at various thresholds, with a higher TPR and lower FPR indicating greater accuracy in the classifier.

**Algorithm 1** Youden's J-statistic

```
1: function FIND_OPTIMAL_THRESHOLD(predictions, labels)
2:     best_threshold ← 0
3:     best_j_statistic ← 0
4:     for threshold ∈ range(0, 1) do
5:         tpr ← CALCULATE_TPR(predictions, labels, threshold)
6:         fpr ← CALCULATE_FPR(predictions, labels, threshold)
7:         j_statistic ← tpr − fpr
8:         if j_statistic > best_j_statistic then
9:             best_j_statistic ← j_statistic
10:            best_threshold ← threshold
11:        end if
12:    end for
13:    return best_threshold
14: end function
```

Figure 5.4: Algorithm to Calculate Youden's J-statistic

Optimizing the threshold for an ROC curve involves identifying the threshold that maximizes the TPR while minimizing the FPR. This is achieved by selecting the threshold that corresponds to the point on the ROC curve closest to the top left corner, where TPR is 1 and FPR is 0. This point is known as Youden's J-statistic, calculated as TPR minus FPR. The optimal threshold is determined by maximizing Youden's J-statistic using gradient descent.

## 5.5 Localisation

In the context of anomaly detection in the Tennessee Eastman process, localisation refers to the ability of the model to not only detect when a fault or anomaly has occurred, but also identify where in the process the fault is happening. This is important because the Tennessee Eastman process is a complex system with many components, and faults can have cascading effects throughout the process.

For example, one of the faults that can occur in the Tennessee Eastman process is the "Reactor Cooling Water Inlet Temperature Step" (IDV(4)). This fault can affect the feature XMV(10), which represents the flow rate of cooling water to the reactor.

Localisation in this context would involve identifying where in the cooling water system the fault is occurring. For instance, the fault could be caused by a problem with the cooling

water inlet temperature sensor, the control valve position, or another component in the cooling water system. By accurately localising the fault, the model can help operators take prompt corrective actions to prevent equipment damage, decreased product yield, or safety hazards caused by the fault.

### 5.5.1 Principal Component Localisation

In the context of anomaly detection in cyber-physical systems, the reconstruction error from the test predictions of an LSTMED model can be used as the data for Principal Component Analysis [27]. Specifically, the test data $\mathbf{X}$ is passed through the LSTMED model to obtain the reconstructed data $\mathbf{X_r}$, and the reconstruction error is computed as $\mathbf{E} = \mathbf{X} - \mathbf{X_r}$. The reconstruction error represents the discrepancy between the original data and its reconstruction, and can be used to identify anomalous samples.

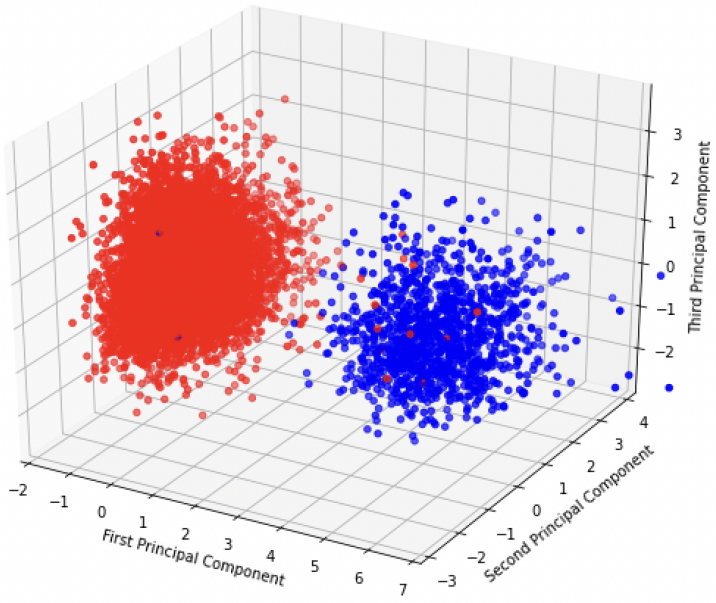



Figure 5.5: Example of PCA Localisation on Reconstruction Error

Performing PCA on the reconstruction error $\mathbf{E}$ allows us to identify the features that contributed the most to the anomaly. As features that caused the anomaly or are related to it are likely to have higher reconstruction error, they will contribute more to the principal components of the data. Therefore, by examining the components of the PCA, we can identify which features contributed the most and are therefore related to the anomaly.

Following the example outlined at the beginning of this chapter, the Reactor Cooling Water Flow is adjusted in order to cause a fault in XMV(10). Using the PCA technique on the reconstruction error we produce a localised list of components to investigate as seen in 5.6.

```
Top k most likely sources of anomaly (k=5)
1: XMV(10)
2: XMV(9)
3: XMEAS(19)
4: XMEAS(9)
5: XMEAS(18)
```

Figure 5.6: Result of localisation using Demonstration (PCA)

### 5.5.2 Threshold Localisation

To localize the cause of an anomaly in time series data from an industrial control system, the reconstruction error for each feature can be compared to the maximum reconstruction error for that feature in normal operation data. A threshold value can be defined for each feature, above which the reconstruction error is indicative of an anomaly [6].If the reconstruction error for a feature in the anomalous data exceeds its threshold value for a significant number of time steps, we can conclude that the feature is likely causing or closely related to the anomaly.

```
Top k most likely sources of anomaly (k=5)
1: XMV(10) with 1145 threshold violations (93.09%)
2: XMEAS(9) with 17 threshold violations (1.38%)
3: XMEAS(29) with 2 threshold violations (0.16%)
4: XMEAS(19) with 1 threshold violations (0.08%)
5: XMEAS(23) with 1 threshold violations (0.08%)
```

Figure 5.7: Result of localisation using Demonstration (Thresholding)

By repeating this process for each feature, we can identify the features that are most likely causing or closely related to the anomaly. However, it is important to note that the most anomalous feature may not necessarily be the root cause of the anomaly due to the interconnectedness [89] of components in an industrial control system. Therefore, a holistic and systematic approach is necessary to identify the root cause of the anomaly.

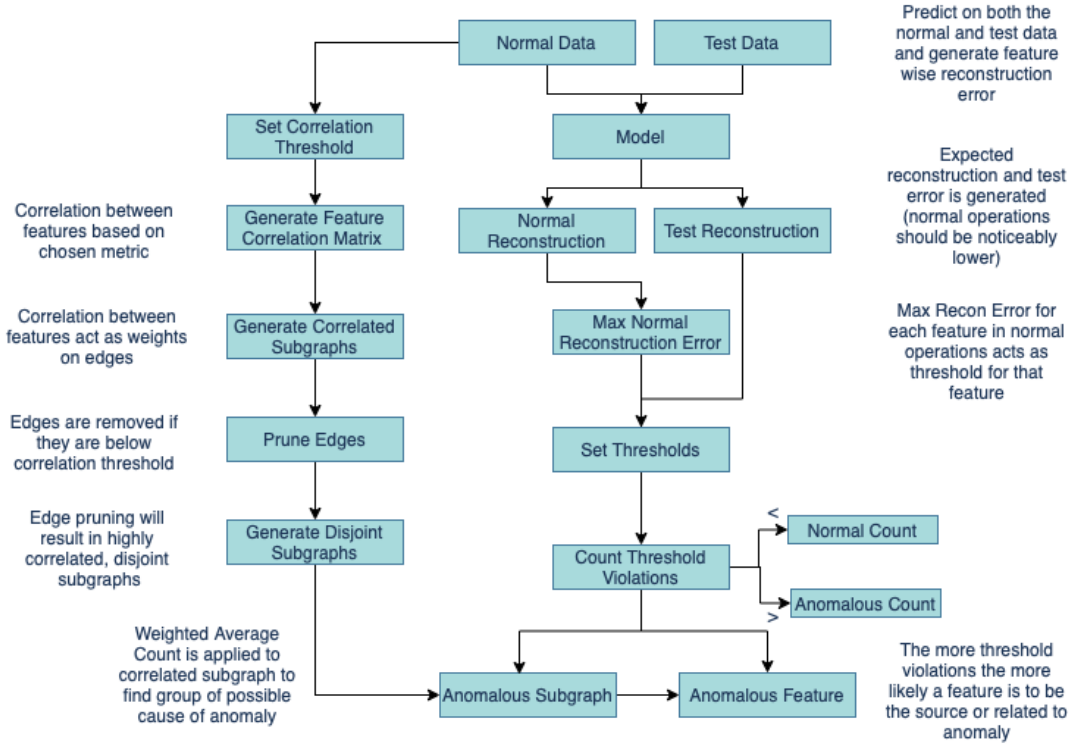



Figure 5.8: CADLAE Architecture: Feature Wise / Subgraph Localisation

### 5.5.3 Correlation Subgraph Localisation

Given a set of features $X = x_1, x_2, ..., x_n$ and a set of data points $D = d_1, d_2, ..., d_m$, the correlation graph $G = (V, E)$ is constructed by calculating the Spearman rank correlation coefficient $\rho_{i,j}$ between all pairs of features $x_i$ and $x_j$, where $i, j \in [1, n]$ and $i \neq j$ [190].

The thresholded correlation graph $G_t = (V, E_t)$ is then obtained by applying a threshold $T$ to the absolute values of the correlation coefficients, such that $E_t = (i, j)||\rho_{i,j}| \geq T$. Each connected subgraph $S_k = (V_k, E_k)$ of $G_t$ represents a set of features that are highly correlated with each other. To localize an anomaly, the average anomaly threshold violations $V(S_k)$ within each subgraph $S_k$ is calculated. The subgraph $S_{max}$ with the highest weight $W(S_k) = V(S_k) \cdot |V_k|$ is identified as the likely cause of the anomaly, where $|V_k|$ denotes the number of vertices in $S_k$ [9].

```
Subgraph C
Rank 1: XMV(10)
Number of Threshold Violations: 1145
Percent of total detected anomalies: 93.09%


Rank 2: XMEAS(9)
Number of Threshold Violations: 17
Percent of total detected anomalies: 1.38%
```

Figure 5.9: Result of localisation using Demonstration (Subgraph)

Localising to a subgraph rather than a single feature is important in CPS because features in the system are likely to be correlated with each other, and identifying a group of correlated features that may be contributing to the anomaly allows for a more comprehensive analysis of the system. By identifying potential interactions between features, a more accurate understanding of the cause of the anomaly can be obtained, leading to better decision making.

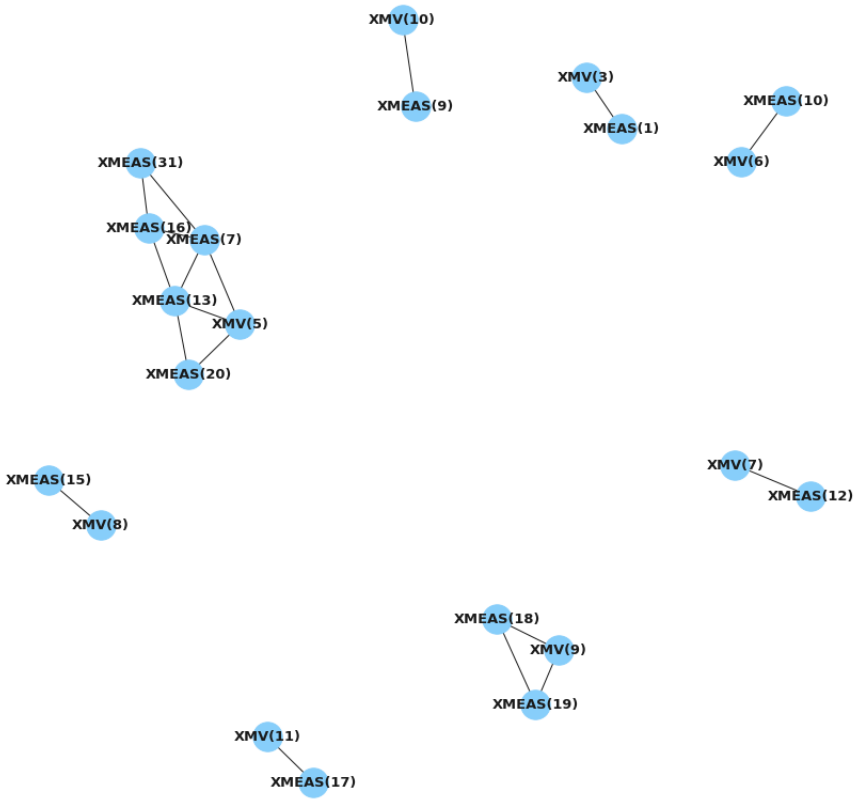



Figure 5.10: Correlation Disconnected Subgraphs

## 5.6 Explanation

### 5.6.1 Tree Based Explanation

The predictions from the proposed unsupervised model are leveraged to act as labels to train a supervised explanation model. The explanation model uses tree-based Gradient Boosting Machines (GBMs) [125] to generate easily interpretable if-then rules.

The human-readable insights provided by the rules allow for quick identification and correction of anomalies in the CPS, minimizing their impact on the system.

Visualized decision trees can be used to identify patterns and correlations in the data

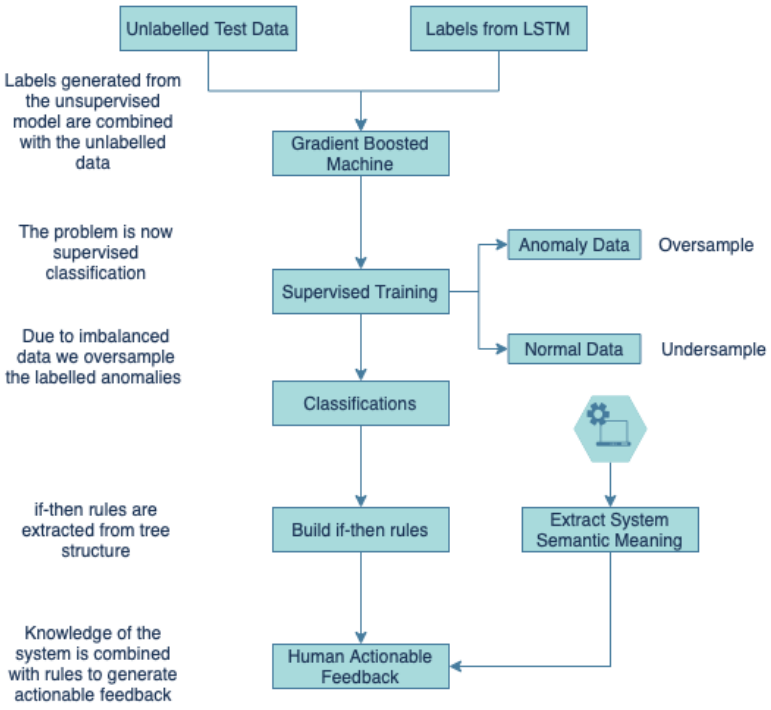



Figure 5.11: CADLAE Architecture: GBM Explainer

associated with anomalies, informing further investigations and improvements to the system.

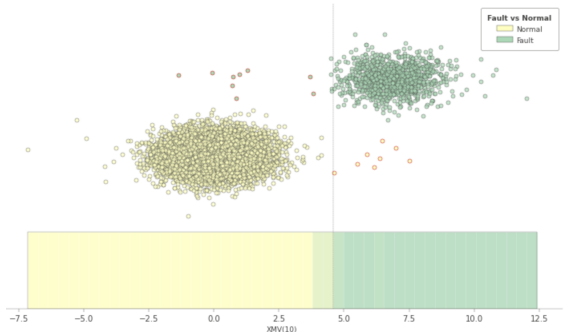



Figure 5.13: GBM Decision Boundary

```
This index was classified as an anomaly. To try resolve this, please take the following actions:

1. Increase Prod Sep Underflow (stream 10) component (XMEAS(14)) to be greater than or equal to 3.18.
2. Increase Component G (stream 11) component (XMEAS(40)) to be greater than or equal to 2.44.
3. Decrease A Feed Flow (stream 1) component (XMV(3)) to be less than -2.73.
4. Decrease Reactor Cooling Water Flow component (XMV(10)) to be less than 4.3.
```

Figure 5.14: Automatically Generated Human Readable Feedback

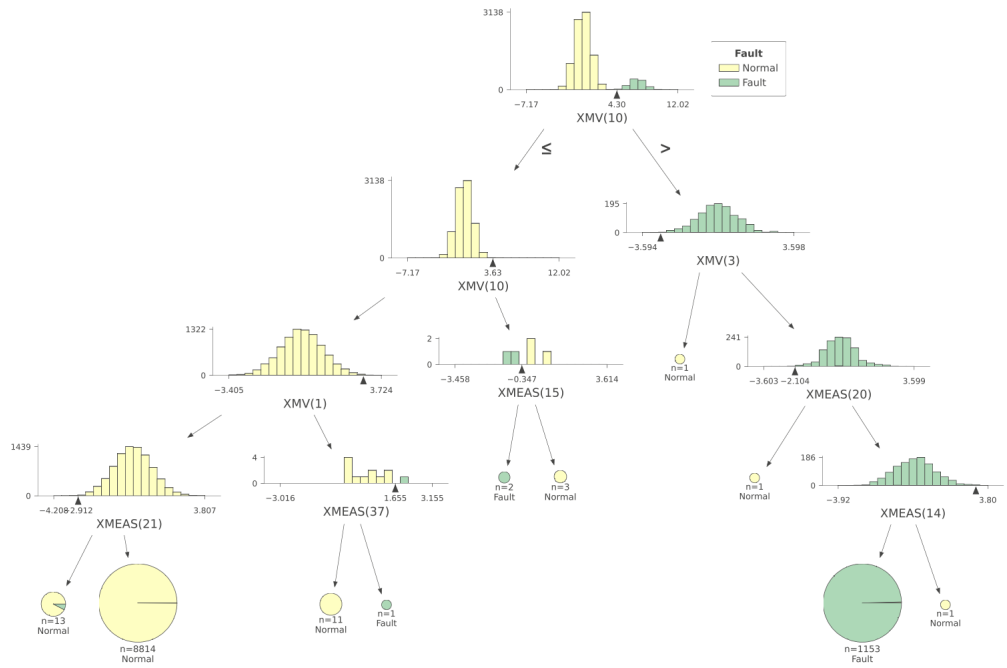



Figure 5.12: Gradient Boosting Machine Global Decision Tree

By increasing the Reactor Cooling Water Flow, we can detect anomalies in the test data, which are then used as labels to create a supervised setting for the GBM to train on. From this, we can generate a global view of how the data was classified, as visualized in the output.

In this particular example, we can see that the change in water flow (increased flow) was detected in XMV(10), as evidenced by the distribution and cutoff seen in the global decision tree Fig 5.12. By parsing the rules and semantics of the Tennessee Eastman, we can generate actionable steps for engineers to correct the fault. These rules as shown in Fig 5.14 are based on the distribution of each feature data and aim to get the system back into normal ranges.

### 5.6.2 Causal Bayesian Network

A causal Bayesian network is a probabilistic graphical model that represents causal relationships between variables in a system. It consists of a directed acyclic graph in which each node represents a variable and each edge represents a causal relationship between variables. The network is parameterized by conditional probability distributions that describe how each variable is influenced by its parent variables.

The model can be expressed mathematically as:

$$P(\boldsymbol{X}, \boldsymbol{Y}) = \prod_{i=1}^{n} P(X_i | Pa_{X_i}) \prod_{i=1}^{m} P(Y_i | Pa_{Y_i}) \tag{5.2}$$

Here, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the sets of observable and unobservable variables, respectively. The conditional probability distributions $P(X_i|Pa_{X_i})$ and $P(Y_i|Pa_{Y_i})$ represent the causal relationships between each variable $X_i$ and $Y_i$ and their respective parent nodes $Pa_{X_i}$ and $Pa_{Y_i}$ in the network [126].

To explain an anomaly in the system, the most probable explanation (MPE) [97] can be computed by finding the assignment of values to the unobserved variables that maximizes the posterior probability:

$$\text{MPE}(\boldsymbol{Y}|\boldsymbol{X}obs) = \arg\max \boldsymbol{Y} P(\boldsymbol{Y}|\boldsymbol{X}_{obs}) \tag{5.3}$$

Here, $\boldsymbol{X}_{obs}$ is the set of observed variables. The MPE provides an explanation for the observed anomaly by identifying the most likely values for the unobserved variables that caused the anomaly, given the observed data [44][95].

Localising the cause of the anomaly in the network involves identifying the variables that have the strongest causal influence on the observed anomaly. This can be done by computing the causal effect of each variable on the observed anomaly using techniques such as the do-calculus. The variables with the strongest causal effect are likely to be the cause of the anomaly and should be investigated further [177].

Building a Bayesian network for a cyber-physical system required a systematic and rigorous approach. The first step was to identify the relevant variables and their dependencies through literature review and examination of the Tennessee Eastman topology [18].

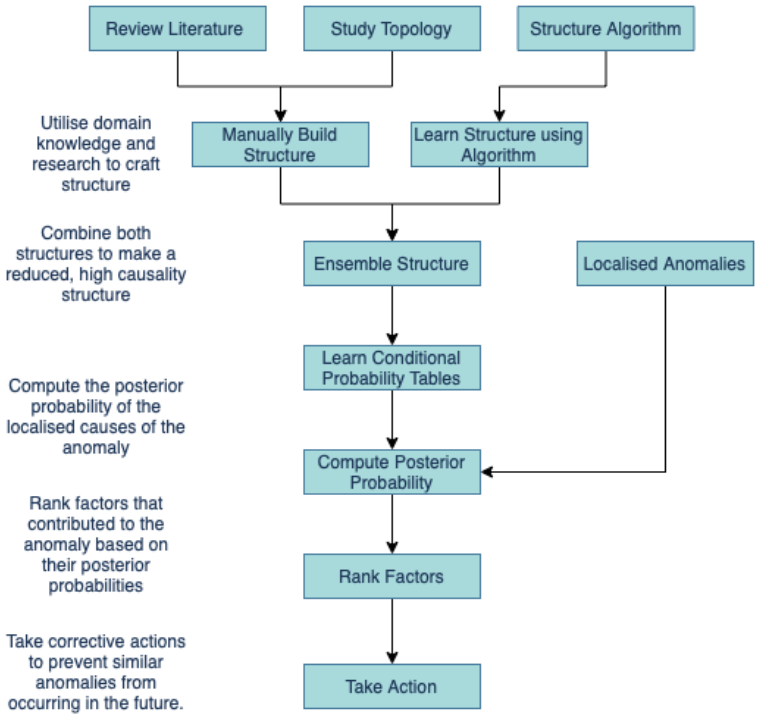



Figure 5.15: CADLAE Architecture: Bayesian Network

The next step was to create a handcrafted Bayesian network by specifying nodes, conditional probability tables, and directed edges. To simplify the network without losing causality [2], the NOTEARS algorithm [179] was used to automatically generate the Bayesian network with a minimum weight threshold. The handcrafted model was then adjusted based on the algorithm output to ensure accurate reflection of the known causal relationships.

Discretization was necessary to work with discrete variables. Once the Bayesian network was constructed, it could be used for inference and to explain the causality of localized anomalies by computing the posterior probability distribution of possible causes [126]. This capability is especially useful in cyber-physical systems to maintain system performance and reliability.

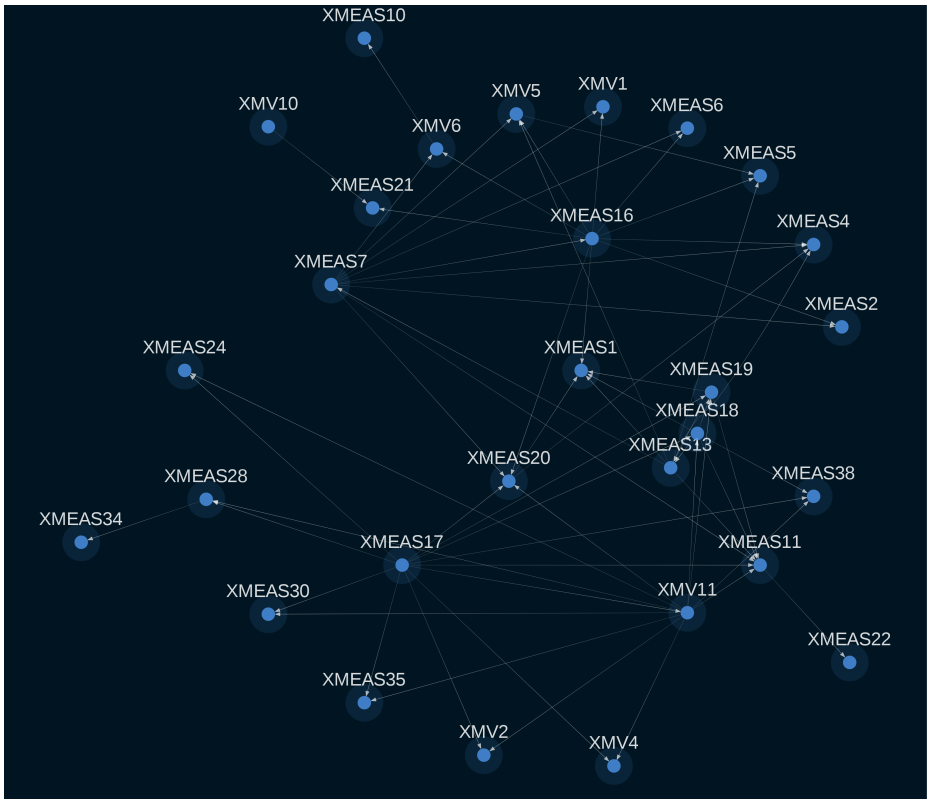



Figure 5.16: Tennessee Eastman Bayesian Network

The original example of XMV(10) was found unsuitable to demonstrate the use of Bayesian networks as it lacks parents in the network. Therefore, XMEAS(10) was used as an alternative example to illustrate the practical applications of Bayesian networks in identifying the probable causes of an anomaly. By examining the parents and grandparents of XMEAS(10) in the Bayesian network, it was discovered that the root cause of the anomaly was XMV(6), as changing its value significantly increased the probability of the anomaly. This highlights the value of using Bayesian networks to model complex systems and trace the causal chain to identify the root cause of anomalies, even when they are not immediately apparent.

$$
\begin{aligned}
P(\text{XMEAS}(10) = \text{Very High} | \text{XMV}(6) = \text{Very High}) &= 0.7 \\
P(\text{XMEAS}(10) = \text{High} | \text{XMV}(6) = \text{Very High}) &= 0.29 \\
P(\text{XMEAS}(10) = \text{Normal} | \text{XMV}(6) = \text{Very High}) &= 0.007 \\
P(\text{XMEAS}(10) = \text{Low} | \text{XMV}(6) = \text{Very High}) &= 0.002 \\
P(\text{XMEAS}(10) = \text{Very Low} | \text{XMV}(6) = \text{Very High}) &= 0.001
\end{aligned}
$$

Figure 5.17: Conditional Probabilities for Explaining TEP Example

### 5.6.3 Causal Explanation vs Correlation Localisation

To provide a clearer understanding of the methods described above, let us consider a concrete example using the TEP system introduced earlier. As shown in Figure 5.18, the subsystem affected in our example is the Reactor subsystem.

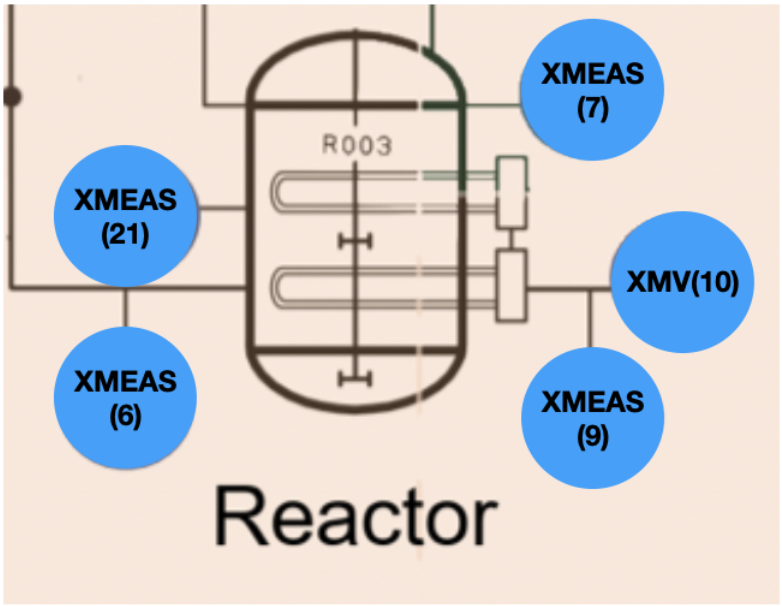



Figure 5.18: Subsystem Effected in Example (Figure 4.9)

Using the Subgraph Correlation method, we first localise the fault to the Reactor subsystem. The within subgraph ranking then indicates that the Reactor Cooling Water Flow (XMV(10)) is the root cause of the anomaly. However, we cannot be certain of this, as the nature of the algorithm only allows us to identify which features are highly related, without providing information on the directionality of their interactions.

On the other hand, using causality techniques such as Bayesian Networks, we can acquire information about how different elements interact and influence each other. In our example, the Bayesian Network in Figure 5.16 shows that XMV(10) has no parents, implying that it has no features influencing it. This information provides a higher level of certainty that the fault is indeed caused by the Reactor Cooling Water Flow element.

Therefore, while the correlation technique can localize the anomaly, it may not provide a definite root cause. In contrast, the causality technique can provide a more certain identification of the root cause by taking into account the directionality of the interactions between the features.

# Chapter 6

# Evaluation and Results

## 6.1 Comparison Models

### 6.1.1 Probabilistic Models

#### Angle-Based Outlier Detection

Kriegel, H.P. and Zimek present a probabilistic approach to detecting outliers in large sets of data called ABOD (Angle-Based Outlier Detection) [93]. The authors argue that existing approaches to outlier detection, which are based on assessing distances in the full-dimensional Euclidean data space, are not suitable for high-dimensional data due to the "curse of dimensionality" [1]. The curse of dimensionality refers to the idea that concepts like proximity, distance, and nearest neighbor become less meaningful as the dimensionality of a data set increases. This is because the relative contrast of the farthest point and the nearest point converges to 0 for increasing dimensionality, making it difficult to discriminate between the nearest and the farthest neighbor in high-dimensional space [176].

To address this problem, the ABOD approach uses a different measure to detect outliers. It assesses the variance in the angles between the difference vectors of a point to other points. This measure is less sensitive to the curse of dimensionality than distance-based measures, making it more suitable for high-dimensional data. Additionally, ABOD does not rely on any parameter selection, which can influence the quality of the achieved ranking.

The authors of the paper present an experimental evaluation of the ABOD approach, comparing it to the well-established distance-based method LOF[23] for various artificial and real-world data sets. They show that ABOD performs especially well on high-dimensional data and that it is more robust than distance-based methods.

Overall, the paper presents a novel approach to outlier detection that addresses the limitations of existing distance-based methods in high-dimensional data, providing a solution that is more suitable for high-dimensional data and does not rely on any parameter selection. This approach can be useful for applications where the identification of different mechanisms responsible for different groups of objects in a data set is important.

#### Empirical-Cumulative-distribution-based Outlier Detection

The paper "ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions" [106] by Zheng Li et al. presents a novel algorithm for unsupervised outlier detection. The authors aim to address the issues of high computational cost, complex hyperparameter tuning [195], and limited interpretability in existing outlier detection methods by proposing a simple yet effective algorithm called ECOD.

Outlier detection refers to the process of identifying data points that deviate from a general data distribution. ECOD is inspired by the fact that outliers are often "rare events" in the tails of a distribution [100][136]. The algorithm first estimates the underlying distribution of the input data in a nonparametric fashion by computing the empirical cumulative distribution per dimension of the data. Then, ECOD uses these empirical distributions to estimate tail probabilities per dimension for each data point and finally computes an outlier score of each data point by aggregating estimated tail probabilities across dimensions.

The key contribution of ECOD is that it is a parameter-free and easy-to-interpret algorithm for unsupervised outlier detection. The authors perform extensive experiments on 30 benchmark datasets and show that ECOD outperforms state-of-the-art outlier detection methods [105][111] in terms of detection performance, runtime efficiency, and interpretability.

One of the strengths of ECOD is that it is simple to understand and implement. The algorithm is based on the idea of computing tail probabilities per dimension, which allows it to effectively capture the rare events that are often outliers. Additionally, ECOD is parameter-free, meaning that it does not require any hyperparameter tuning, which can be time-consuming and difficult to perform.

However, one of the limitations of ECOD is that it may not perform well with datasets that have multiple distributions or non-uniform distributions. In these cases, the algorithm may struggle to effectively capture the underlying distribution of the data and may result in false positive or false negative outlier detection.

### 6.1.2 Change Point

#### Pruned Exact Linear Time

The authors in [85] present the Pruned Exact Linear Time (PELT) method. Changepoint detection is the process of identifying points in a time series where the underlying probability distribution of the data changes. This problem is commonly approached by minimizing a cost function over possible numbers and locations of changepoints.

The PELT method is an exact method, meaning that it guarantees to find the true changepoints in the data, under mild conditions. The computational cost of the PELT method is linear in the number of observations, denoted as $O(n)$, where $n$ is the number of data points. This is a significant improvement over existing methods for the same problem, such as Binary Segmentation and Segment Neighbourhood, which have computational costs of $O(n \log n)$ and $O(Qn^2)$ respectively, where $Q$ is the maximum number of changepoints to be searched for. In scenarios where the number of changepoints increases linearly with $n$, this can correspond to a computational cost that is cubic in the length of the data.

The PELT method is based on the Optimal Partitioning approach of [76], but involves a pruning step within the dynamic program. This pruning step reduces the computational cost of the method, but does not affect the exactness of the resulting segmentation. The PELT method can be applied to find changepoints under a range of statistical criteria, such as penalized likelihood and quasi-likelihood.

In the simulation studies presented in the paper, the authors compare the performance of PELT with existing methods, such as Binary Segmentation [155] and Segment Neighbourhood [10]. They show that PELT can be orders of magnitude faster than these alternative exact methods, and also demonstrate that the exactness of the PELT approach can lead to substantial improvements in the accuracy of the inferred segmentation of the data.

#### Bottom Up

Here the authors propose the use of bottom-up change point detection for signal segmentation[73]. This method, referred to as BottomUp, is a fast and efficient technique for signal segmentation as it starts with many change points and successively deletes the less significant ones. Unlike binary segmentation, which is a greedy procedure, bottom-up segmentation is a generous approach. The signal is first divided into many sub-signals along a regular grid. Then, contiguous segments are successively merged according to a measure of how similar they are. This method has been previously analyzed in literature such as [83] and [55].

The benefits of bottom-up segmentation include its low complexity, which is of the order of $O(n)$, where $n$ is the number of samples. Additionally, it can extend any single change point detection method to detect multiple change points, making it suitable for a wider range of applications. Furthermore, it can work whether the number of regimes is known beforehand or not, providing a robust solution to signal segmentation.

In comparison to other change point detection methods, BottomUp has a number of advantages. For example, it has a lower computational complexity than other methods such as the Segment Neighbourhood. Additionally, it can provide more accurate results than the Binary Segmentation algorithm.

### 6.1.3 Proximity Based

#### K-Nearest Neighbours

In this paper, the authors propose a formulation for distance-based outliers detection. They define an outlier as a point that has a large distance from its nearest neighbor [140]. They rank each point based on its distance to its nearest neighbor and declare the top points in this ranking as outliers. To find these outliers, the authors propose several algorithms including a classical nested-loop join algorithm, an index join algorithm, and a highly efficient partition-based algorithm. The partition-based algorithm first partitions the input data set into disjoint subsets and then prunes entire partitions as soon as it is determined that they cannot contain outliers, resulting in substantial savings in computation.

The authors also mention that the problem of detecting outliers has been extensively studied in the statistics community, with traditional approaches requiring knowledge of the

underlying data distribution. The authors propose a distance-based definition for outliers that is simple and intuitive and doesn't require any prior knowledge of data distributions. The proposed definition and algorithms are general enough to model statistical outlier tests for normal, poisson and other distributions.

One advantage of the proposed method is its ease of use and intuition. The distance-based definition of outliers is straightforward and doesn't require extensive prior knowledge of the data. The partition-based algorithm is also highly efficient, resulting in substantial savings in computation.

However, there are also some limitations to this method,the ranking of outliers may not always reflect the true underlying structure of the data, and further analysis may be necessary to confirm the presence of outliers.

### 6.1.4 Graph Based

#### Learnable Unified Neighbourhood-based Anomaly Ranking

The paper "LUNAR: Unifying Local Outlier Detection Methods via Graph Neural Networks" [59] by Goodge et al. presents a novel approach for anomaly detection by unifying the local outlier methods through a graph neural network. The authors propose the method LUNAR that aims to address the limitations of existing local outlier detection methods, such as LOF and DBSCAN, by making them learnable [151].

Local outlier methods are popular for their simple principles and strong performance in unstructured, feature-based data. However, these methods have a lack of trainable parameters, making them unable to adapt to a particular set of data. In this paper, the authors demonstrate that local outlier methods can be viewed as a specific case of the more general message passing framework used in graph neural networks. This allows the authors to introduce learnability into local outlier methods in the form of a neural network.

LUNAR uses information from the nearest neighbours of each node to find anomalies in a trainable way. The method learns to propagate information through the graph in a manner that enables it to capture both local and global information. The authors show that LUNAR performs significantly better than existing local outlier methods and state-of-the-art deep baselines. Moreover, the performance of the method is found to be more robust to different settings of the local neighbourhood size.

The main advantage of LUNAR is its ability to learn from the data, which enables it to adapt to specific datasets and overcome the limitations of traditional local outlier methods. However, the method may be computationally expensive and may not perform well on datasets with high dimensional features.

### 6.1.5 Deep Learning

#### Adversarially Learned Anomaly Detection

The paper "Adversarially Learned Anomaly Detection" [191] by Zenati et al. introduce a new method for anomaly detection based on Generative Adversarial Networks (GANs)[37].

The proposed method, Adversarially Learned Anomaly Detection (ALAD), is based on bi-directional GANs and uses adversarially learned features for the anomaly detection task. The bi-directional GAN consists of a generator and a discriminator, where the

generator creates samples from a random noise vector, and the discriminator determines whether the sample is real or generated. The adversarial training between the generator and the discriminator results in a more robust and discriminative representation of the data.

ALAD uses reconstruction errors based on the adversarially learned features to determine if a data sample is anomalous. The reconstruction error measures the difference between the input data and its reconstructed representation, and if the difference is high, it indicates that the data sample is anomalous. The authors build on recent advances to ensure data-space and latent-space cycle-consistencies and stabilize GAN training, which results in significantly improved anomaly detection performance.

Formally, the BiGAN[46] and AliGAN[49] models match the joint distribution $p_G(x, z) = p_Z(z)p_G(x|z)$ and $p_E(x, z) = p_X(x)p_E(z|x)$ with an adversarial discriminator network $D_{xz}$ that takes $x$ and $z$ as inputs. The BiGAN and AliGAN determine the discriminator $D_{xz}$ that separates the joint distributions $p_G(x, z)$ and $p_E(x, z)$ and uses it for anomaly detection by computing the reconstruction error as $||G(E(x)) - x||_1$.

The authors evaluate ALAD on a range of image and tabular datasets and show that it achieves state-of-the-art performance in terms of accuracy and runtime. ALAD is faster at test time than the only published GAN-based method, making it a promising alternative for real-world anomaly detection applications.

ALAD is a promising alternative to traditional anomaly detection methods, especially for complex and high-dimensional data. However, it is important to note that GANs are known to be challenging to train, and the stability and robustness of the training process are crucial factors in the success of ALAD.

### Beta Variational Autoencoders

The authors of this article are proposing a new perspective on the emergence of disentangled representations in Variational Autoencoders (VAEs) [24], specifically in the beta-VAE variant. They take a rate-distortion theory perspective and show the conditions under which representations that align with the underlying generative factors of the data will emerge as the VAE is being optimized.

In the VAE framework, the goal is to learn the joint distribution of the data their latent generative factors. The VAE objective is typically defined as the Evidence Lower Bound (ELBO) which is given by the following formula [21]:

$$ELBO = Eq\phi(z|x)[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{6.1}$$

where $q_\phi(z|x)$ is the approximate posterior distribution of the latent variables given the data and $p_\theta(x|z)$ is the generative model. $D_{KL}$ is the Kullback-Leibler divergence [175] which measures the difference between two probability distributions.

The beta-VAE variant adds an extra hyperparameter, beta, to the VAE objective. The idea is that by constricting the effective encoding capacity of the latent bottleneck and encouraging the latent representation to be more factorised, the model will learn disentangled representations.

The authors propose a modification to the training regime of beta-VAE that increases the information capacity of the latent code during training, which they argue will facilitate the robust learning of disentangled representations in beta-VAE without the trade-off in

reconstruction accuracy that is typically seen. The goal of the research is to gain a better theoretical understanding of how beta-VAE works in order to scale disentangled factor learning to more complex datasets.

### Deep Support Vector Data Description

This paper introduces a new anomaly detection method called Deep Support Vector Data Description (Deep SVDD) [148] which is a deep learning-based approach to anomaly detection.

Deep SVDD is a novel approach to deep anomaly detection inspired by kernel-based one-class classification [153] and minimum volume estimation. The method trains a neural network by minimizing the volume of a hypersphere that encloses the network representations of the data. The idea is that by forcing the network to extract the common factors of variation and closely map the data points to the center of the sphere, it will be able to detect deviations from this description as anomalies.

To train the network, the authors minimize the following objective function:

$$L = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{c}|^2 - \frac{1}{\nu n} Tr(\boldsymbol{K}) + \frac{1}{\nu} \tag{6.2}$$

where L is the objective function, $\boldsymbol{x}_i$ is the input data, $\boldsymbol{c}$ is the center of the sphere, $\nu$ is a trade-off hyperparameter and Tr(K) is the trace of the kernel matrix.

The authors demonstrate the effectiveness of their method on MNIST and CIFAR-10 image benchmark datasets as well as on the detection of adversarial examples of GTSRB stop signs. They also provide theoretical properties of their method, including robustness to adversarial examples and resistance to overfitting in the presence of small amounts of anomalous data. They also claim that the method is an effective way to detect anomalies in high-dimensional, data-rich scenarios, and it is more efficient than other classical algorithms such as One-Class SVM or Kernel Density Estimation which often fail in these scenarios.

## 6.2 Evaluation Metrics

In this paper, several evaluation metrics were used to assess the performance of the deep learning model for anomaly detection. These metrics include accuracy, precision, recall, F1 score, and ROC-AUC. Each metric provides a unique perspective on the model's performance and together they provide a comprehensive evaluation of the model's ability to accurately identify anomalies.

### Accuracy

The accuracy [187] is defined as the ratio of true positives (TP) and true negatives (TN) to the total number of instances in the dataset:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.3}$$

where TP, TN, FP, and FN stand for true positives, true negatives, false positives and false negatives respectively. These terms are used in the context of binary classification problems, where the model is trying to classify instances into one of two classes (e.g. anomaly or non-anomaly). TP refers to instances that are correctly classified as positive (anomaly), TN refers to instances that are correctly classified as negative (non-anomaly), FP refers to instances that are incorrectly classified as positive (false alarm), and FN refers to instances that are incorrectly classified as negative (missed anomaly).

Accuracy is a simple metric that provides an overall assessment of the model's performance, however, it can be misleading when the dataset is imbalanced. This means that when the number of positive instances is low compared to the number of negative instances, a model that simply predicts negative for all instances will have a high accuracy. Therefore, it is important to look at other metrics that provide a more nuanced evaluation of the model's performance.

### Precision

Precision [169], defined as the ratio of true positives to true positives plus false positives:

$$Precision = \frac{TP}{TP + FP} \tag{6.4}$$

It measures the proportion of true positives among the positive predictions made by the model. This metric is important when it is more important to have a low false alarm rate, i.e. when the cost of identifying a normal instance as an anomaly is high.

### Recall

Recall [169], defined as the ratio of true positives to true positives plus false negatives:

$$Recall = \frac{TP}{TP + FN} \tag{6.5}$$

It measures the proportion of true positives among all actual positive instances in the dataset. This metric is important when it is more important to have a high detection rate, i.e. when the cost of missing an anomaly is high.

### F1 Score

F1 score [110], defined as the harmonic mean of precision and recall:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.6}$$

It is a balance of precision and recall, and it is commonly used when you want to balance precision and recall trade-off.

### ROC-AUC

ROC-AUC, Receiver Operating Characteristic - Area Under the Curve, is a widely used performance measure for binary classification problems. It plots the true positive rate (TPR) against the false positive rate (FPR), where TPR is defined as

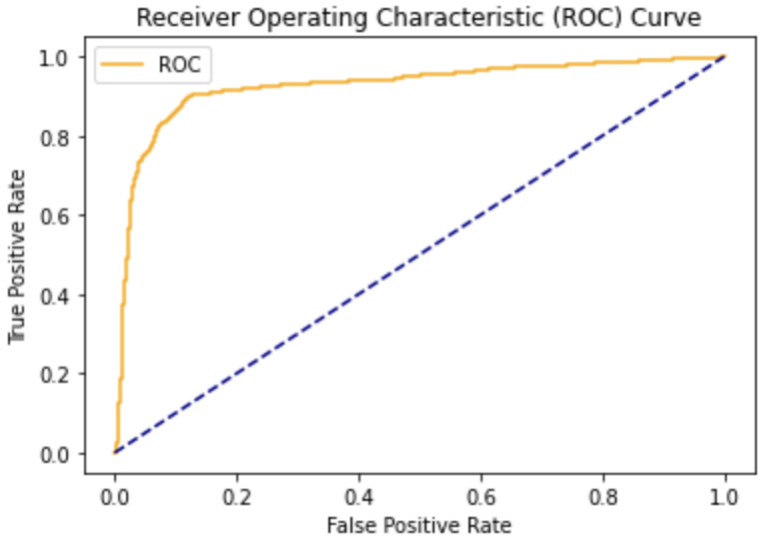



Figure 6.1: ROC-AUC Plot for Fault F

$$TPR = \frac{TP}{TP + FN} \tag{6.7}$$

and FPR is defined as

$$FPR = \frac{FP}{FP + TN} \tag{6.8}$$

AUC (Area Under the Curve) [71] is a measure of the area under the ROC curve, and it ranges from 0 to 1, where a higher value indicates better performance. This metric is insensitive to class imbalance and it can be used as a threshold-independent metric. In summary, the use of these evaluation metrics in anomaly detection provides a comprehensive evaluation of the model's performance. They provide detailed information about the model's ability to accurately identify positive instances while minimizing the number of false positives. Each metric provides a unique perspective on the model's performance and they are chosen based on the specific needs of the problem and the cost of false alarms and missed anomalies. Accuracy is a simple overall assessment of the model's performance but can be misleading when the dataset is imbalanced. Precision and recall provide a more nuanced evaluation, precision measures the proportion of true positives among the positive predictions made by the model, while recall measures the proportion of true positives among all actual positive instances. F1 score is a balance of precision and recall and is used when balancing precision and recall trade-off is desired. ROC-AUC is a widely used performance measure for binary classification problems, insensitive to class imbalance and can be used as a threshold-independent metric. Together these metrics provide a robust evaluation of the model's performance and a comprehensive understanding of the model's ability to accurately identify anomalies.

## 6.3 Results

### 6.3.1 Single Fault Data - Tennessee Eastman

The following section outlines the findings from the classification experiments conducted on the single fault test data of the Tennessee Eastman process. This dataset consists of multiple instances of the same fault occurring over time, and the objective of the experiments was to assess the system's ability to identify and diagnose a particular type of fault.

Despite the repetitive nature of the faults, the single fault test data remains a challenging dataset. It mimics scenarios in real-world systems where a single fault, such as a faulty temperature sensor in a chemical reactor, can cause the system to repeatedly detect and diagnose the same type of fault.

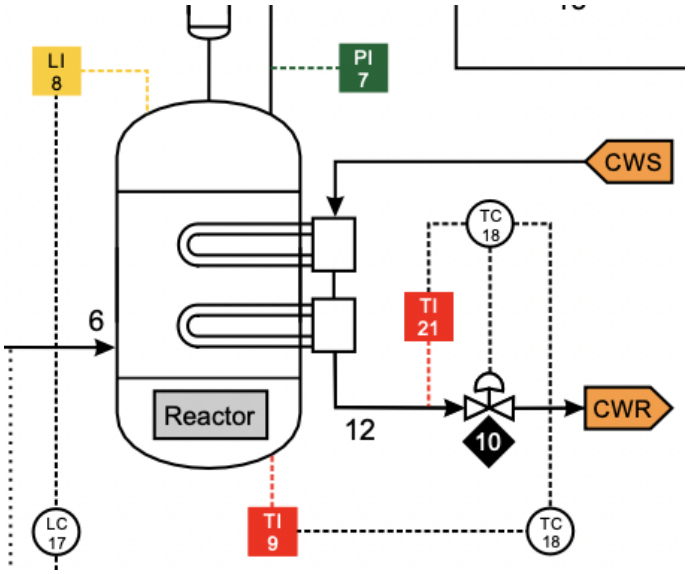



Figure 6.2: Demonstration of Single Fault

To provide an illustrative example of a single fault within the Tennessee Eastman simulation, we shall examine the case of Fault C in Table **??**. This particular fault pertains to the manipulation of the temperature of the cooling water being supplied to the reactor. In practical terms, this means that the flow rate of the reactor cooling water XMV(10) (labelled 10 in Fig 6.2) is increased. As a result, the temperature inside the reactor decreases, and this change can be detected by monitoring the reactor temperature with the XMEAS(9) sensor (labelled TI 9 in Fig 6.2).

The effect of this fault is significant, as it results in an alteration of the flow rate of the cooling water, specifically an increase in XMV(10). This increase in flow rate, in turn, has a notable effect on the temperature within the reactor, which is carefully monitored by the reactor temperature sensor XMEAS(9).

As this scenario is for single fault, after the fault is introduced and its effects are observed, the system is restored to its normal state. The same fault is then introduced at a different interval, and this process is repeated multiple times to create a dataset for

that specific fault. This process is repeated for all the faults listed in Table **??**.

| Model | Accuracy | Precision | Recall | F1 Score | ROC-AUC |
|---|---|---|---|---|---|
| ABOD | 82.67% | 85.02% | 95.50% | 86.93% | 74.89% |
| ALAD | 78.57% | 85.87% | 89.86% | 87.15% | 54.31% |
| BottomUp | 57.16% | 57.11% | 91.14% | 70.22% | 57.33% |
| DeepSVDD | 78.86% | 78.55% | 97.24% | 84.29% | 79.88% |
| ECOD | 82.71% | 88.97% | 91.50% | 90.10% | 61.89% |
| KNN | 82.52% | 84.23% | 96.05% | 86.75% | 76.83% |
| LUNAR | 77.24% | 78.52% | 95.50% | 83.43% | 72.97% |
| PELT | 77.15% | 81.62% | 90.03% | 83.63% | 62.32% |
| VAE | 75.92% | 76.93% | 95.51% | 82.46% | 72.56% |
| **Proposed Model** | **91.59%** | **92.23%** | **98.17%** | **95.03%** | **89.47%** |

Table 6.1: Results Table for Single Fault Data

In the table above, the results of several anomaly detection models are presented, along with their performance metrics. The proposed model demonstrates the highest overall performance, with an accuracy of 91.59%, precision of 92.23%, recall of 98.17%, F1 score of 95.03%, and ROC-AUC of 89.47%. This indicates that the proposed model has a high ability to correctly identify anomalous instances while also minimizing the number of false positives.

The other models also show generally good performance, with accuracy scores ranging from 57.16% to 82.71%. However, when considering the precision, recall, and F1 score, the proposed model stands out as having the highest values, indicating a strong balance between correctly identifying anomalous instances and minimizing false positives.

It is also worth noting that some models, such as ALAD and ECOD, have lower ROC-AUC scores compared to the other models. The ROC-AUC metric measures the ability of a model to distinguish between positive and negative classes, so lower scores in this metric may indicate that these models have difficulty differentiating between normal and anomalous instances.

In summary, the proposed model shows the highest overall performance in terms of accuracy, precision, recall, F1 score and ROC-AUC, indicating its strong ability to identify anomalous instances while maintaining a low number of false positives. While other models also show good performance, their metrics are not as high as the proposed model.

The bar chart provides a clear visual representation of the results of several anomaly detection models. This chart allows us to easily compare and contrast the performance of the different models based on the F1 Score metric. As we can see from the chart, the proposed model stands out as having the highest F1 score of 95.03%. This is a very impressive result and indicates that the proposed model has a strong ability to accurately identify anomalous instances while maintaining a good balance between precision and recall.

It is also worth noting that the other models also show relatively good performance, but none of them are able to achieve a F1 score as high as the proposed model. This is a clear indication that the proposed model has a significant advantage in terms of performance.

The confusion matrix for the proposed model, as shown in Fig 6.4, provides detailed information about the model's performance in terms of true positive (TP), false positive

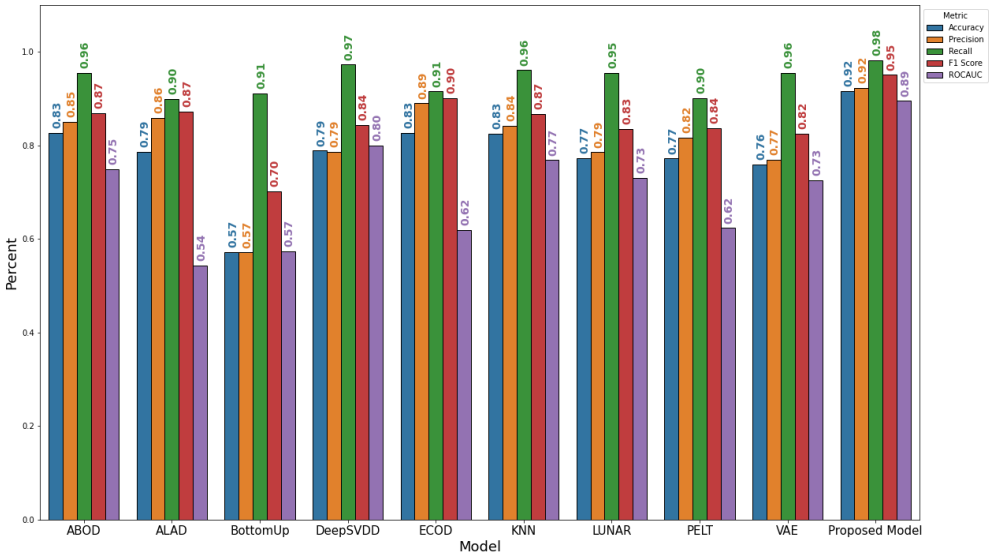



Figure 6.3: Visualisation of Single Fault Results

(FP), true negative (TN), and false negative (FN) rates.

The matrix shows that the proposed model has a true positive rate of 83.24%, meaning that 83.24% of the anomalous instances were correctly identified by the model. It also has a true negative rate of 91.45%, meaning that 91.45% of the normal instances were correctly identified by the model.

Additionally, the matrix shows that the proposed model has a false positive rate of 8.55%, meaning that 8.55% of the normal instances were incorrectly identified as anomalous by the model. And it has a false negative rate of 16.76%, meaning that 16.76% of the anomalous instances were incorrectly identified as normal by the model.

Confusion matrices for the other models can be found in Appendix A.1. These matrices can be used to further analyze the performance of the other models and compare them to the proposed model.

### 6.3.2 Multi Fault Data - Tennessee Eastman

The results of the classification experiments on the multiple fault data are presented in this section. The data set in question encompasses multiple faults that occur over time, each of which is of a different type and can overlap with one another. The objective of these experiments is to assess the ability of the system to detect and diagnose multiple faults simultaneously.

The multiple fault data was generated through a process that simulates a real-world scenario in which multiple faults can occur within a system over time. The process takes into consideration various factors such as the type of faults, the frequency of faults, and the timing of faults in order to create a complex and dynamic environment for the anomaly

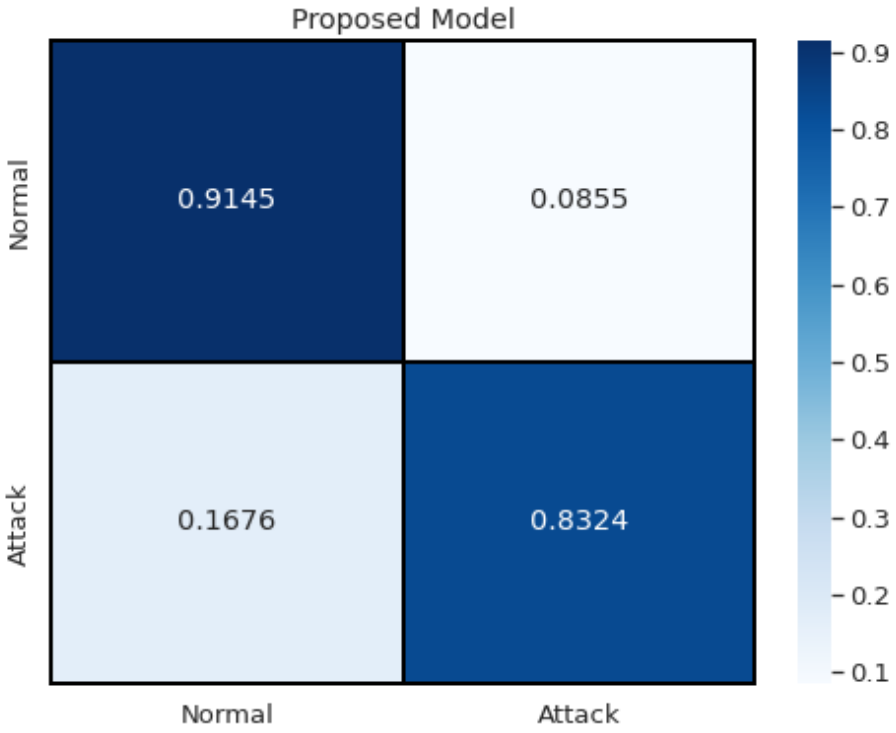


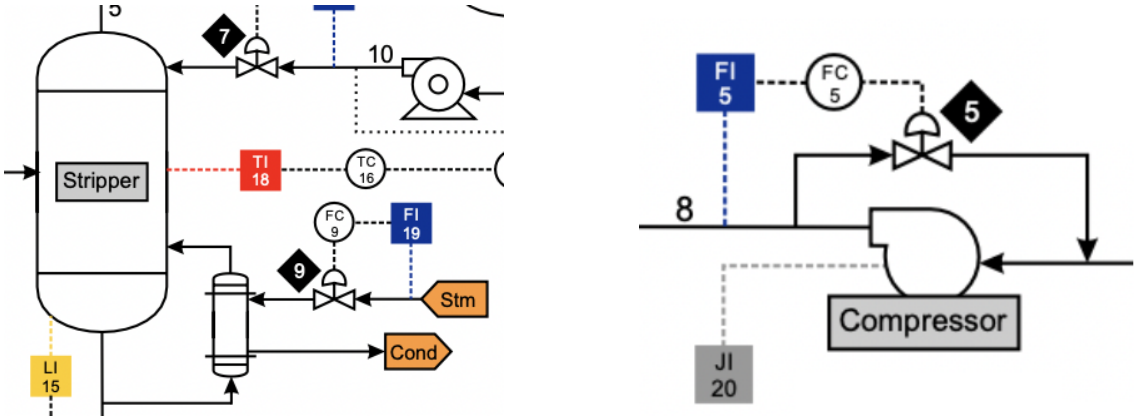

Figure 6.4: Proposed Model Confusion Matrix - Single Fault

detection models to operate within. An example that utilises the schematics and semantics of the Tennessee Eastman is process is provided in Fig 6.5.

In this multi-fault scenario, we are examining two simultaneous faults that are occurring in different subsystems of the system. The first fault involves the Stripper subsystem, which is a complex system consisting of multiple sensors that monitor Stripper level, pressure, temperature, and steam flow. When the Stripper steam valve (XMV(9)) is manipulated as shown in Fig 6.5a, it causes an increase in pressure (XMEAS(16)) and steam flow (XMEAS(19)) readings. The second fault involves the Compressed Recycle valve (XMV(5)) as shown in Fig 6.5b, which is in a different subsystem, and its manipulation is reflected in the Compressor Work sensor (XMEAS(20)).

The challenge with this multi-fault scenario is that these two faults are unrelated, and they are happening in an overlapping time interval in separate subsystems. This makes it more difficult to detect and localize the faults because there is increased noise and potentially conflicting information from different subsystems. The models must have the ability to accurately detect and classify multiple faults at the same time. The overlapping and intermingled nature of the faults can make it difficult for the models to generalize and learn from the data, potentially leading to false positive or false negative results.

The results of the multiple fault classification experiments are presented in a table showing the performance of various anomaly detection techniques. The table provides a




(a) Compressed Recycle valve (%)

(b) Stripper Value Fault (%)

Figure 6.5: Multi Fault TEP Example

| Model | Accuracy | Precision | Recall | F1 Score | ROC-AUC |
|---|---|---|---|---|---|
| ABOD | 81.53% | 81.78% | 97.84% | 89.09% | 80.17% |
| ALAD | 86.44% | 92.95% | 92.38% | 92.67% | 51.04% |
| BottomUp | 82.43% | 93.41% | 88.43% | 93.86% | 50.28% |
| DeepSVDD | 80.76% | 80.61% | 98.21% | 88.54% | 81.61% |
| ECOD | 89.28% | 92.97% | 95.3% | 94.12% | 69.29% |
| KNN | 81.32% | 81.5% | 97.88% | 88.94% | 80.29% |
| LUNAR | **90.9%** | **94.94%** | 95.18% | **95.06%** | 68.94% |
| PELT | 84.22% | 94.31% | 89.23% | 89.56% | 50.70% |
| VAE | 83.93% | 84.88% | 97.36% | 90.69% | 78.78% |
| **Proposed Model** | 90.22% | 90.29% | **99.01%** | 94.45% | **89.81%** |

Table 6.2: Results Table for Multi Fault Data

comparison of the accuracy, precision, recall, F1 score, and ROC-AUC of each model.

The proposed model outperforms the other models in terms of recall, achieving a score of 99.01%. This indicates that the proposed model is able to correctly identify a high proportion of the multiple faults present in the data set. Additionally, the proposed model also has a high accuracy score of 90.22% and an F1 score of 94.45%. These metrics reflect the balance between the model's ability to correctly identify faults and minimize false alarms.

The ROC-AUC of the proposed model is 89.81%, which is also the highest among all models. This metric evaluates the ability of the model to differentiate between normal and anomalous instances. A higher ROC-AUC indicates that the proposed model is able to effectively distinguish between multiple faults in the data set.

In terms of F1 score, LUNAR had the highest score of 95.06%, however, its recall score was lower compared to the proposed model. This shows that while LUNAR was able to identify a high proportion of faults, it also had a higher rate of false negatives, which could result in missed faults.

In Fig 6.6, it can be concluded that many of comparison models have a higher precision

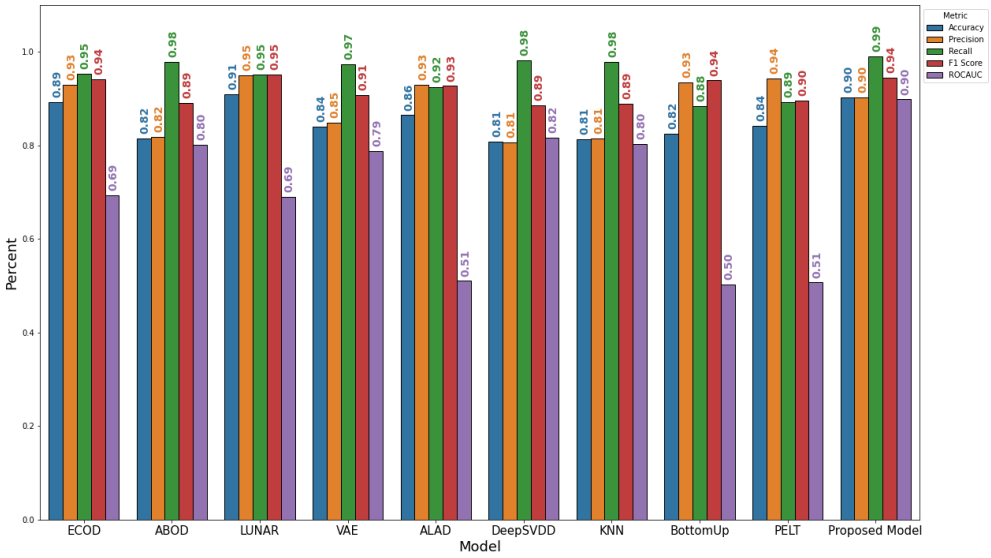



Figure 6.6: Visualisation of Multi Fault Results

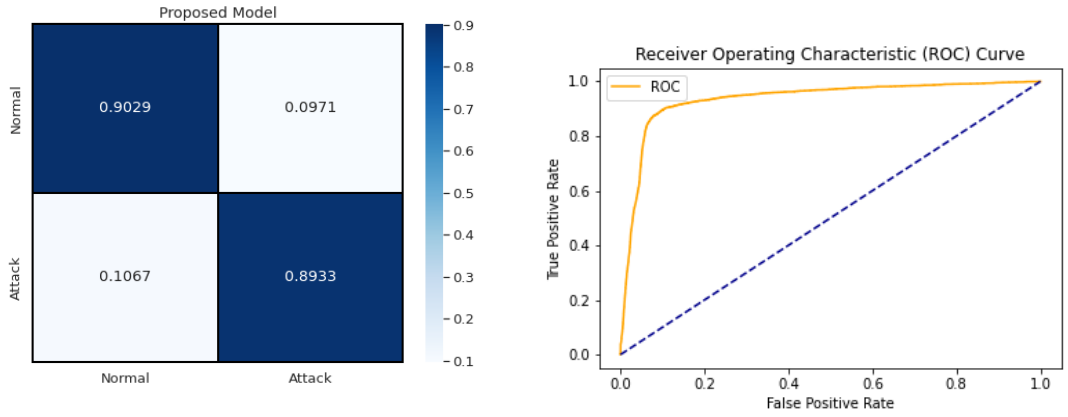
but lower recall, while the proposed model has a lower precision but higher recall. This trade-off between precision and recall is common in many machine learning models and is often dependent on the specific problem being solved and the cost of false positives vs false negatives.

It is important to consider both precision and recall when evaluating the performance of a model, as a high precision or recall in isolation may not be sufficient for certain applications. In some cases, a high precision may be desired, while in others a high recall may be more important.

The results show a trade-off between precision and recall in the proposed and comparison models, and the better AUC of the proposed model indicates its overall better performance considering both precision and recall.

### 6.3.3 Additional Datasets

In this section, results from the evaluation of various models on additional benchmark datasets are presented. The aim of incorporating these benchmark datasets was to reduce any potential bias in the evaluation of the models, particularly the proposed model, as the data used for its training and evaluation was generated internally. This additional evaluation provides a more comprehensive and objective assessment of the performance of the models. The benchmark datasets were taken from the paper "ADBench: Anomaly Detection Benchmark" by Han et al. This paper presents a comprehensive evaluation of 30 anomaly detection algorithms on 57 benchmark datasets, referred to as ADBench [65], with the aim of providing meaningful insights into the role of supervision and anomaly types in algorithm performance. They found that different anomaly detection algorithms




(a) Confusion Matrix (b) ROC-AUC Plot

Figure 6.7: Proposed Model Charts for Multi Fault Data

are better suited to different types of anomalies. The ADBench benchmark provides a comprehensive and fair evaluation of existing anomaly detection algorithms and is open-source and fully reproducible, making it useful for future researchers in algorithm selection and design.

| Dataset | Model | Performance Metrics | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Precision | Recall | F1 Score | ROC-AUC |
| Cover | ECOD | **92.44%** | 92.79% | 99.53% | **96.04%** | **89.02%** |
| | ABOD | 20.94% | 20.09% | **99.99%** | 33.46% | 59.99% |
| | LUNAR | 46% | 45.43% | 99.98% | 62.47% | 72.24% |
| | $\beta$-VAE | 84.71% | 84.72% | 99.79% | 91.64% | 84.19% |
| | ALAD | 93.89% | **94.57%** | 99.22% | 96.84% | 62.74% |
| | DeepSVDD | 83.33% | 83.76% | 99.28% | 90.86% | 63.69% |
| | KNN | 38.84% | 38.19% | **99.99%** | 55.27% | 68.93% |
| | Proposed | 84.99% | 84.93% | 99.88% | 91.8% | 87.84% |
| Shuttle | ECOD | 97.93% | 98.09% | 99.65% | 98.87% | 97.04% |
| | ABOD | 92.69% | 92.06% | **99.98%** | 95.87% | 96.03% |
| | LUNAR | 86.76% | 85.62% | 99.45% | 92.25% | 92.81% |
| | $\beta$-VAE | 92.38% | 91.83% | 99.88% | 95.69% | 95.29% |
| | ALAD | 88.20% | 87.63% | 99.50% | 93.19% | 91.25% |
| | DeepSVDD | 83.28% | 81.89% | 99.94% | 90.02% | 90.65% |
| | KNN | 92.41% | 91.76% | 99.91% | 95.70% | 95.88% |
| | Proposed | **99.45%** | **99.64%** | 99.76% | **99.70%** | **98.42%** |
| Thyroid | ECOD | 95.43% | 95.73% | 99.56% | 97.61% | 90.34% |
| | ABOD | 91.16% | 90.98% | **99.93%** | 95.24% | **94.41%** |
| | LUNAR | 94.37% | 94.43% | 99.78% | 97.03% | 93.45% |

| | | | | | | |
|---|---|---|---|---|---|---|
| | $\beta$-VAE | 93.17% | 93.10% | 99.87% | 96.36% | 94.40% |
| | ALAD | 84.89% | 87.16% | 97.00% | 91.82% | 45.73% |
| | DeepSVDD | **97.17%** | **99.30%** | 97.82% | **98.56%** | 90.40% |
| | KNN | 90.07% | 89.85% | **99.93%** | 94.63% | 93.85% |
| | Proposed | 92.72% | 92.67% | 99.84% | 96.12% | 93.65% |
| Satimage-2 | ECOD | 96.69% | 96.84% | 99.80% | 98.3% | 91.38% |
| | ABOD | 41.32% | 40.51% | 99.34% | 57.66% | 70.25% |
| | LUNAR | 65.13% | 64.65% | 99.76% | 78.53% | 82.33% |
| | $\beta$-VAE | 76.22% | 75.95% | 99.92% | 86.3% | 85.86% |
| | ALAD | 95.5% | 96.76% | 98.65% | 97.7% | 80.49% |
| | DeepSVDD | 86.87% | 86.78% | 99.89% | 92.87% | 89.87% |
| | KNN | 54.57% | 53.94% | 99.81% | 70.08% | 76.97% |
| | Proposed | **99.23%** | **99.24%** | **99.98%** | **99.61%** | **98.92%** |
| Mammography | ECOD | 95.57% | 96.45% | 98.97% | 97.7% | 79.38% |
| | ABOD | **97.42%** | **99.32%** | 97.42% | **98.69%** | 86.87% |
| | LUNAR | 87.66% | 88.18% | 99.05% | 93.3% | 78.13% |
| | $\beta$-VAE | 93.87% | 94.91% | 98.75% | 96.79% | 74.76% |
| | ALAD | 89.23% | 91.48% | 97.3% | 94.3% | 83.86% |
| | DeepSVDD | 97.39% | **99.97%** | 97.42% | 98.68% | **89.98%** |
| | KNN | 94.26% | 95.25% | 98.81% | 97% | 76.09% |
| | Proposed | 85.22% | 85.4% | **99.34%** | 91.84% | 81.93% |
| Campaign | ECOD | 83.2% | 89.53% | 90.8% | 90.16% | 67.08% |
| | ABOD | **83.37%** | 88.99% | 91.43% | **90.19%** | 69.06% |
| | LUNAR | 80.22% | **93.08%** | 85.26% | 89% | 47.43% |
| | $\beta$-VAE | 82.26% | 87.95% | 91.1% | 89.5% | 67.77% |
| | ALAD | 77.07% | 87.89% | 85.77% | 86.82% | 49.46% |
| | DeepSVDD | 77.43% | 84.27% | 88.88% | 86.52% | 59.97% |
| | KNN | 82.63% | 87.86% | 91.58% | 89.68% | 69.29% |
| | Proposed | 77.48% | 79.72% | **93.08%** | 85.88% | **71.77%** |
| Annthyroid | ECOD | 89.1% | **94.35%** | 93.81% | 94.08% | 62.53% |
| | ABOD | 85.08% | 84.11% | 99.51% | 91.18% | 89.99% |
| | LUNAR | **91.6%** | 93.46% | 97.29% | **95.33%** | 82.22% |
| | $\beta$-VAE | 91.2% | 92.9% | 97.39% | 95.09% | 82.59% |
| | ALAD | 76.77% | 81.37% | 92.42% | 86.54% | 53.51% |
| | DeepSVDD | 88.46% | 94.33% | 93.17% | 93.75% | 58.68% |
| | KNN | 84.98% | 83.84% | 99.76% | 91.11% | **90.8%** |
| | Proposed | 79.98% | 78.54% | **99.55%** | 87.81% | 87.3% |

Table 6.3: Benchmark Datasets Results

The results show that the proposed model achieved good results on the benchmark datasets, with some of the best results among the compared models in terms of accuracy, precision, recall, F1 score, and ROC-AUC. However, it is also important to note that the performance of the models may vary depending on the specific dataset, and the choice of model should be based on the specific requirements and characteristics of the dataset in question.

| Dataset | # Samples | # Features | Anomaly | Domain |
|---|---|---|---|---|
| Cover [19] | 286048 | 10 | 0.96% | Botany |
| Shuttle [143] | 49097 | 9 | 7.15% | Astronautics |
| Thyroid [139] | 3772 | 6 | 2.47% | Healthcare |
| Satimage-2 [143] | 5803 | 36 | 1.22% | Astronautics |
| Mammography [181] | 11183 | 6 | 2.32% | Healthcare |
| Campaign [130] | 41188 | 62 | 11.27% | Finance |
| Annthyroid [138] | 7200 | 6 | 7.42% | Healthcare |

Table 6.4: Benchmark Datasets Overview

### 6.3.4 Localisation

This section presents the results of the localization experiments on the single fault test data. The dataset contains numerous anomalies of varying duration for a specific fault type, and the objective of these experiments is to evaluate the system's ability to localize the faults to a correlated subgraph and a specific component within the cyber physical system. The proposed deep learning approach was applied to perform both types of localization and the results are shown below.

| Fault | Number of Faults | Subgraph | Correct Subgraph | Component | Correct Component |
|---|---|---|---|---|---|
| Fault C | 15 | Subgraph C | 15 | XMV(10) | 15 |
| Fault H | 15 | Subgraph C | 15 | XMV(10) | 15 |
| Fault K | 15 | Subgraph H | 12 | XMEAS(19) | 9 |
| Fault G | 15 | Subgraph H | 13 | XMEAS(18) | 12 |
| Fault N | 15 | Subgraph B | 11 | XMV(5) | 11 |
| **Accuracy** | | | **88%** | | **82.67%** |

Table 6.5: Fault Localisation Results

The table 6.5 presents the results of the localization experiments on the single fault test data. The data contains 15 anomalies for each of the five faults, Fault C, H, K, G, and N, with the objective of evaluating the system's ability to localize the faults to a correlated subgraph and a specific component within the cyber physical system.

For the first two faults, Fault C and H, the system accurately localized the faults to the correct subgraph, Subgraph C, 100% of the time. In both cases, the system also correctly identified the component responsible for the fault, XMV(10), with 100% accuracy.

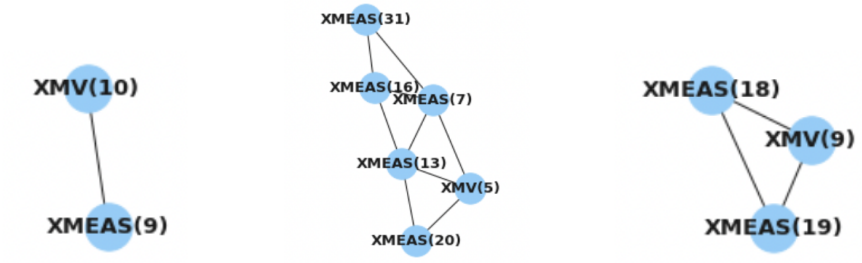
For Fault K, the system correctly localized the fault to the correct subgraph, Subgraph H, 80% of the time. In terms of component identification, the system correctly identified XMEAS(19) as the responsible component 60% of the time.

Similarly, for Fault G, the system correctly localized the fault to the correct subgraph, Subgraph H, 86.67% of the time. The component identification accuracy was 80% for this fault, with XMEAS(18) being correctly identified as the responsible component.

Finally, for Fault N, the system correctly localized the fault to the correct subgraph, Subgraph B, 73.33% of the time. The component identification accuracy was 73.33% for

this fault, with XMV(5) being correctly identified as the responsible component.

The overall accuracy for subgraph localization was 88%, while the overall accuracy for component identification was 82.67%. These results demonstrate the effectiveness of the proposed deep learning approach in localizing anomalies in CPS, although there is clear room for improvement in terms of identification accuracy.




(a) Subgraph A (b) Subgraph B (c) Subgraph H

Figure 6.8: Subgraphs displayed in Table 6.5

## 6.4 Hyper Parameter Tuning

Hyperparameter tuning is an essential step in the development of a machine learning model as the choice of hyperparameters can greatly impact the model's performance. The process of hyperparameter tuning involves optimizing the hyperparameters of a model in order to improve its performance on the training data.

In this paper, the hyperparameter tuning was performed using the Tree-structured Parzen Estimator (TPE) algorithm [17] which is implemented in the hyperopt package [16]. The TPE algorithm is a Sequential Model-Based Optimization (SMBO) [74] approach that models the distribution of the hyperparameters given the model's performance, $p(x|y)$.

The TPE algorithm transforms the configuration space, which is described by a graph-structured generative process, into non-parametric densities. This is done by replacing the distributions of the configuration prior with non-parametric densities such as truncated Gaussian mixtures for uniform variables, exponentiated truncated Gaussian mixtures for log-uniform variables, and re-weighted categorical distributions for categorical variables.

The TPE algorithm defines the distribution of the hyperparameters given the model's performance, $p(x|y)$, using two densities: $(x)$ and $g(x)$. $(x)$ is the density formed using the observations of hyperparameters that correspond to lower model performance (loss), while $g(x)$ is the density formed using the remaining observations. The TPE algorithm chooses a threshold value, $y^*$, such that a quantile $\gamma$ of the observed losses is less than $y^*$. The TPE algorithm uses a sorted list of observed variables to keep track of the densities, allowing for linear scaling in the number of observed variables.

In summary, the TPE algorithm models the distribution of the hyperparameters given the model's performance, $p(x|y)$, using non-parametric densities to estimate the distribution. The algorithm chooses a threshold value, $y^*$, to divide the observations into two

densities, $(x)$ and $g(x)$ and maintains a sorted list of observed variables for efficient computation.

### 6.4.1 Loss Function Comparison

#### Cosine similarity loss

Given two vectors $\mathbf{u}$ and $\mathbf{v}$, the cosine similarity between them is defined as:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}|_2 |\mathbf{v}|_2}. \tag{6.9}$$

This similarity measure can be used to define a cosine similarity loss between two sets of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathbf{y}_1, \ldots, \mathbf{y}_n$ as:

$$\mathcal{L}_{\cos}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}n) = \sum i = 1^n (1 - \text{sim}(\mathbf{x}_i, \mathbf{y}_i)). \tag{6.10}$$

This loss function encourages the model to learn embeddings that are similar for pairs of vectors that are semantically related, and dissimilar for pairs of vectors that are not related [69].

#### One-Class SVM (OCSVM) loss

Given a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the OCSVM algorithm learns a hyperplane that separates the data points from the origin in a high-dimensional space. The objective function of the OCSVM algorithm can be written as:

$$\min_{\mathbf{w}, \boldsymbol{\xi}, \rho} \frac{1}{2} |\mathbf{w}|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho, \tag{6.11}$$

subject to:

$$\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n, \tag{6.12}$$

where $\boldsymbol{\phi}(\mathbf{x}_i) = [\phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i)]$ is a feature map that transforms the input $\mathbf{x}_i$ into a high-dimensional space, and $\phi_n(\cdot)$ is a kernel function. The parameter $\nu$ controls the fraction of the training data that is allowed to be misclassified by the hyperplane.

The OCSVM objective function can be used as a loss function for training a model on a dataset with known outliers, by setting $\nu$ to a small value and treating the inliers as the positive class and the outliers as the negative class. The model learns a decision function that assigns high scores to inliers and low scores to outliers [183].

#### Attention-Based loss

$$\mathcal{L}_{\text{att}}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}_n, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}n) = \sum i = 1^n \boldsymbol{\alpha}_i \cdot \text{CE}(\mathbf{y}_i, \hat{\mathbf{y}}_i), \tag{6.13}$$

where $\text{CE}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ is the cross-entropy loss between the ground truth label $\mathbf{y}_i$ and the predicted label $\hat{\mathbf{y}}_i$, and $\boldsymbol{\alpha}_i$ is the attention weight assigned to the input vector $\mathbf{x}_i$.

The attention weights are computed as a softmax function over the dot product between the query vector $\mathbf{q}$ and the key vectors $\mathbf{k}_1, \ldots, \mathbf{k}_n$, which are derived from the input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$:

$$\boldsymbol{\alpha}_i = \frac{\exp(\text{sim}(\mathbf{q}, \mathbf{k}i))}{\sum j = 1^n \exp(\text{sim}(\mathbf{q}, \mathbf{k}_j))}. \tag{6.14}$$

The query vector $\mathbf{q}$ is typically a learnable parameter of the model [116].

**Mean squared error (MSE) loss**

$$\mathcal{L}_{\text{MSE}}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}n) = \frac{1}{n} \sum i = 1^n |\mathbf{y}_i - \hat{\mathbf{y}}_i|^2, \tag{6.15}$$

where $\mathbf{y}_i$ is the ground truth label and $\hat{\mathbf{y}}_i$ is the predicted label for the input vector $\mathbf{x}_i$. The model learns to minimize the average squared difference between the predicted and ground truth labels [86].

**Kullback-Leibler (KL) divergence**

The KL divergence is a measure of how different two probability distributions are. For two discrete distributions $P$ and $Q$ over the same set of outcomes, the KL divergence is defined as:

$$\text{KL}(P|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \tag{6.16}$$

In the context of machine learning, the KL divergence is often used as a loss function to train generative models to approximate a target distribution. The objective function for training such models can be written as:

$$\min_\theta \text{KL}(P_{\text{data}}|P_\theta), \tag{6.17}$$

where $\text{KL}(P_{\text{data}}|P_\theta)$ is the Kullback-Leibler (KL) divergence between the data distribution $P_{\text{data}}$ and the model distribution $P_\theta$.

The KL divergence is a measure of the difference between two probability distributions $P$ and $Q$, and is defined as:

$$\text{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \tag{6.18}$$

In the case of training a generative model, $P_{\text{data}}$ is the true data distribution and $P_\theta$ is the model distribution parameterized by the model's parameters $\theta$. The goal is to minimize the KL divergence between the two distributions, which can be seen as minimizing the difference between the true data distribution and the model's distribution. This is equivalent to maximizing the likelihood of the data under the model:

$$\max_\theta \log P_\theta(\mathbf{x}) = \max_\theta \log \frac{P_\theta(\mathbf{x})}{P_{\text{data}}(\mathbf{x})} - \text{KL}(P_{\text{data}}||P_\theta) \tag{6.19}$$

The first term in the equation above is the log-likelihood of the data under the model, and the second term is the KL divergence between the data distribution and the model distribution. By maximizing the log-likelihood and minimizing the KL divergence, we are simultaneously trying to match the data distribution and generate high-quality samples from the model [171].

**Results**

The proposed model is trained using these loss functions, and their performance is evaluated based on several metrics, including accuracy, precision, recall, F1-score, and ROC curve. The results provide insights into the effectiveness of different loss functions for detecting anomalies in industrial processes and guide the development of more accurate and reliable anomaly detection models.

| Loss Function | Accuracy | Precision | Recall | F1 Score | ROC-AUC |
|---|---|---|---|---|---|
| Cosine Similarity | 90.07% | 90.62% | 98.49% | 94.39% | 87.08% |
| One-Class SVM | **91.34%** | **91.81%** | 98.71% | **95.13%** | 88.78% |
| Attention-Based | 90.67% | 91.15% | 98.58% | 94.72% | 87.76% |
| Mean Squared Error | 91.21% | 91.62% | **98.76%** | 95.05% | **88.96%** |
| Kullback-Leibler | 90.56% | 90.91% | **98.76%** | 94.67% | 87.73% |

Table 6.6: Results Table for Loss Functions

# Chapter 7

# Discussion

## 7.1 Model Performance

### 7.1.1 Tennessee Eastman Single Fault

The proposed model outperformed all other models in terms of accuracy, precision, recall, F1 score, and ROC-AUC [70]. The superior performance of the proposed LSTM model could be attributed to its ability to model the temporal dependencies and correlations in the data. Cyber-physical systems generate sequential data that has complex temporal patterns, and the LSTM model is specifically designed to handle such data. The model uses a combination of input, output, and forget gates to selectively store or discard information from previous time steps, allowing it to capture long-term dependencies in the data. In contrast, other models like ABOD [93], ALAD[191], and KNN[140] do not consider the temporal nature of the data, making them less effective for detecting anomalies in dynamic systems.

Furthermore, the hyperparameter tuning process could have contributed to the suboptimal performance of the other models. Anomaly detection algorithms typically require the selection of multiple hyperparameters, and finding the optimal combination of hyperparameters can be a challenging task. For example, algorithms like ABOD, ALAD, and KNN need to balance for different types of anomalies, which can require setting different hyperparameters for different scenarios. However, this balancing act can lead to suboptimal performance, especially when dealing with complex data like cyber-physical systems. In contrast, the proposed LSTM model requires minimal hyperparameter tuning, which simplifies the training process and reduces the risk of suboptimal performance. It is important to note that the Tennessee Eastman data used in this study was self-generated, which may introduce a potential source of bias. There is a possibility that the dataset was tailored to complement the performance of the proposed model, which could have inflated its results.

Overall, the proposed LSTM model outperforms the other models in this study because of its ability to model the temporal dependencies and correlations in the data and its reduced sensitivity to hyperparameter tuning. The model's strong performance suggests that it could be an effective tool for detecting anomalies in cyber-physical systems, particularly those that generate sequential data.

### 7.1.2 Tennessee Eastman Multi Fault

The results of the proposed LSTM encoder-decoder model are promising, with an accuracy of 90.22%, precision of 90.29%, recall of 99.01%, true F1 score of 94.45%, and ROC-AUC of 89.81%. These findings suggest that the LSTM model is effective at detecting anomalies in cyber-physical systems.

However, it's worth noting that the results for multi-fault type data are more balanced compared to single-fault type data discussed earlier, with the LSTM model being good but not as dominant. It is possible this difference is due to the hyperparameter tuning being more effective for other models when dealing with multi-fault data that contains different types of anomalies. This result highlights the importance of carefully selecting and tuning the model hyperparameters to achieve balanced results when dealing with multi-fault data. The other models, such as ABOD, DeepSVDD[148], and LUNAR[59], were able to achieve higher results for a variety of metrics compared to the LSTM model for multi-fault data, indicating that the best model is dependent on which metric is valued the most.

However, the ROC-AUC score of the proposed model is the highest among all models, indicating that it is better at distinguishing between normal and anomalous data points. Overall, the results suggest that the proposed LSTM model can be a valuable tool for anomaly detection in cyber-physical systems. However, careful consideration of the dataset configuration and selection of appropriate hyperparameters is important for achieving balanced and effective results.

### 7.1.3 Benchmark Datasets

The performance of LSTM model was evaluated using several benchmark datasets that are commonly used in the field of anomaly detection [66][149][115][40]. These datasets were selected to ensure fairness in the evaluation, as they are not self-generated and thus are less biased compared to other datasets that were used. The LSTM model showed strong performance across most of the datasets, which is an indication of its effectiveness in detecting anomalies.

However, it is worth noting that the performance of the LSTM model was not consistent across all the datasets. Some of the datasets were non-temporal, which means that they do not have a time-series structure, and therefore, the LSTM model may not be the most suitable approach. The variability in performance across the datasets highlights the importance of selecting appropriate models that are tailored to the specific characteristics of the data.

Despite the variability in performance across the datasets, the LSTM model demonstrated promise in the domain of anomaly detection for cyber physical systems. The results suggest that the model has the potential to be an effective tool for detecting anomalies in complex systems. However, further research is needed to explore its performance in different scenarios and to identify the best approaches for optimizing its performance in specific use cases.

### 7.1.4 Overall performance

One of the key strengths of the proposed model is its ability to perform well across all datasets, including self-generated and normal benchmark datasets. This indicates that

the model's performance is due to its ability to learn the underlying patterns of the data and not to the data being tailored to the model. This is a significant advantage over other models that require carefully curated data to perform well.

Another advantage of the Seq-2-Seq LSTM Encoder Decoder model is that it requires very little hyperparameter tuning to be effective. This means that the model can be applied to new datasets with minimal adjustments, making it easy to use straight away with new data. The ability of the model to generalize well to new data is essential in practical applications, where labeled data may be scarce or unavailable. The Seq-2-Seq LSTM Encoder Decoder model is also effective in detecting anomalies in both temporal and non-temporal data. In cyber physical systems, temporal data is the norm, and the model's ability to detect anomalies in this type of data is essential. However, the model's ability to detect anomalies in non-temporal data indicates its usefulness in other areas outside of cyber physical systems. This is particularly relevant in industries such as finance and healthcare, where the data is often non-temporal. The model's unsupervised nature is another advantage in practical applications, where labeled data is often scarce or unavailable. Unsupervised models can learn patterns from the data without the need for labels, making them a cost-effective solution for anomaly detection in cyber physical systems. However, the model requires a significant amount of data to be effective, which can be a problem in some circumstances. In some applications, such as medical diagnosis, it may not be feasible to collect large amounts of data. This can limit the effectiveness of the Seq-2-Seq LSTM Encoder Decoder model in such applications. Another potential limitation of the model is its long training time. The Seq-2-Seq LSTM Encoder Decoder model requires a significant amount of training data, and training the model can take a long time. This can be a limiting factor in practical applications, where real-time detection is required. In conclusion, the Seq-2-Seq LSTM Encoder Decoder model is a promising solution for anomaly detection in cyber physical systems. The model's ability to learn patterns from the data and detect anomalies based on deviations from this pattern is a significant advantage over other models. However, the model's limitations, such as the need for a significant amount of data and long training time, need to be addressed to make it more practical for real-world applications. Future research can focus on optimizing the model's training process, developing more efficient architectures that require less data, and exploring the potential of combining unsupervised and supervised learning methods to improve the accuracy of anomaly detection in cyber physical systems.

## 7.2 Limitations of Study

### 7.2.1 Limitations with Data Used

The reliance of the study on internally generated data presents a significant limitation for anomaly detection in cyber-physical systems. While this approach enables researchers to reduce anomalies and exercise greater control over the experimental setting, it also exposes the study to the risk of selectively choosing data. Specifically, the use of in-house data may lead to a scenario where researchers cherry-pick data that supports their hypothesis while ignoring data that contradicts or weakens their findings [122].

This potential for cherry-picking data is of particular concern in the study since it may significantly compromise the validity and reliability of the conclusions. Analysis of

a smaller subset of data may fail to accurately represent the true relationship or pattern observed in the larger dataset, leading to erroneous or biased conclusions with far-reaching implications for detecting anomalies in cyber-physical systems.

To mitigate this issue, additional datasets from sources other than the study were incorporated, thus enhancing the sample size and reducing the possibility of cherry-picking. This approach not only improved the robustness and generalizability of the findings but also allowed for a more comprehensive account of anomaly detection in cyber-physical systems.

### 7.2.2 Limitations in Anomaly Detection Approach

The focus in this research was primarily on achieving high accuracy in detecting anomalies. However, the speed at which a model detects an anomaly is equally important, as it can have significant implications for mitigating the effects of the anomaly [99].

Anomalies in cyber-physical systems can have severe consequences, and the ability to detect them quickly is crucial for minimizing their impact. For instance, in a power grid, a delay in detecting an anomaly could result in widespread power outages or equipment damage, leading to significant financial losses or even endangering human lives.

To gain a more comprehensive understanding of the performance of the anomaly detection model, it would be useful to measure its speed in addition to its accuracy. This could help assess its suitability for different real-world scenarios.

One limitation of the this is that the anomalies were not quantified, and all anomalies were treated equally. However, in real-world situations, different types of anomalies can have varying effects on the system. It may be useful to prioritize the detection of certain types of anomalies over others.

For example, in a power grid, detecting voltage spikes with higher accuracy may be more critical than detecting voltage dips. Similarly, in a manufacturing process, prioritizing the detection of faulty sensor readings over detecting process delays may be beneficial [165].

Future studies could explore the effectiveness of different anomaly detection models in detecting specific types of anomalies and compare their performance against general anomaly detection models. Such studies could help identify the most critical types of anomalies for different systems and develop targeted anomaly detection approaches that prioritize the detection of these anomalies.

### 7.2.3 Limitations in Localisation

One key limitation of the paper is that the localisation techniques used were heavily dependent on the reconstruction error of the LSTM encoder-decoder model. This means that if the model failed to detect anomalies accurately, it would also propagate to the localization techniques, resulting in potentially inaccurate localization of the anomaly. This creates a single point of failure in the model, which may limit its effectiveness in real-world scenarios.

In cyber-physical systems, accurate localization of anomalies is crucial for identifying the root cause of the problem and taking appropriate corrective actions. Therefore, any

limitations in the localization techniques could lead to significant consequences, including system failures or safety hazards.

To address this limitation, future research could explore alternative localization techniques that are less dependent on the reconstruction error of the LSTM encoder-decoder model. For example, incorporating additional sensor data or using multiple models in parallel could improve the accuracy and robustness of the localization techniques.

Additionally, it is essential to evaluate the effectiveness of the model and its localization techniques under different conditions, such as varying levels of noise or different types of anomalies. This would help to identify the strengths and weaknesses of the model and provide insights into its suitability for different real-world scenarios.

### Thresholding localisation limitations

One limitation of the thresholding localization technique used is that the effectiveness of the thresholds is highly dependent on the consistency of normal operations. If the normal process is more volatile, with significant variations in sensor readings or system behavior, it can make the thresholding technique less effective.

In cyber-physical systems, normal operations can vary depending on several factors, such as environmental conditions or user demand. If the thresholding technique is designed based on historical data that does not accurately represent the current system's behavior, it can lead to false positive or false negative results, which can impact the accuracy and reliability of the anomaly detection system.

To mitigate this limitation, future research could explore alternative thresholding techniques that can adapt to changing system behavior and incorporate real-time data analysis to update the thresholds. Additionally, incorporating machine learning models that can learn from the system's behavior and automatically adjust the thresholds could improve the accuracy and robustness of the thresholding technique.

Another approach is to combine the thresholding technique with other localization techniques to reduce the risk of false positive or false negative results. For example, using a combination of thresholding and clustering techniques could improve the accuracy and effectiveness of the localization process.

### PCA localisation limitations

A limitation of the PCA localization technique used is that it takes the variables with the largest contribution to the first eigenvector, which explains the most variance. While this technique explained a significant amount of the variance in our data, in cases where a high number of principal components are required, these features may not actually contribute that much to the overall variance.

In other words, the PCA localization technique assumes that the variables with the highest contribution to the first eigenvector are the most important for explaining the underlying patterns in the data. However, this assumption may not always hold, and other variables may also play a crucial role in detecting anomalies.

To mitigate this limitation, future research could explore alternative feature selection techniques that can identify the most relevant variables for detecting anomalies in cyber-physical systems. For example, using machine learning models that can learn from the

data and identify the most important features for anomaly detection could enhance the accuracy and effectiveness of the PCA localization technique.

### 7.2.4 Limitations in Explainability

#### Bayesian Network Structure

An important limitation of the Bayesian network discussed is the reliance on manual inference and structure learning algorithms to define the network structure. This approach was necessary due to the lack of a detailed description of the system's topology and interactions between components.

However, the accuracy of the inferred network structure may be limited by the quality and completeness of the data sources used for inference, as well as the assumptions and biases of the researcher. Additionally, the inferred network structure may not capture all the important interactions and dependencies between components, leading to a potentially incomplete or inaccurate representation of the system.

This limitation can also impact the interpretability and explanatory power of the model. The inferred network structure may not provide a comprehensive understanding of how different components of the system are related, which can make it difficult to explain the model's outputs or identify potential sources of anomalies.

To address this limitation, future research could explore alternative approaches for inferring network structure, such as using expert knowledge or leveraging additional data sources to improve the accuracy and completeness of the network representation. Additionally, researchers could consider using alternative modeling techniques that may be more robust to incomplete or uncertain data, such as fuzzy logic or neural networks.

#### Bayesian Network Structure Complexity

Another limitation of the Bayesian network used is its complexity and computational expense. Building and running the network, as well as conducting what-if scenario analysis, can require significant computational resources and expertise. This may limit the applicability of the model in practical settings, where real-time detection and response to anomalies is critical.

Furthermore, the computational complexity of the Bayesian network may also impact its scalability to larger and more complex systems. As the number of components and interactions in the system increases, the size and complexity of the network also increases, leading to a corresponding increase in computational resources required for analysis.

#### GBM Limitation

A key limitation of the Gradient Boosting Machines (GBM) explainer technique used is that it is a supervised technique, which requires labeled data to train the model. However, in our case, the labeled data was not available, so we used the predicted labels as the pseudo ground truth. This approach has the potential to introduce errors and inaccuracies, which can have a significant negative impact on the structure of the tree and the rules derived from it.

Inaccurate labels can lead to biased or misleading results, which can undermine the validity and reliability of the model. Moreover, inaccuracies can also lead to the overfitting or underfitting of the model, which can further degrade its performance.

## 7.3 Deep Learning vs Traditional Techniques

Deep learning has emerged as a powerful tool for detecting, localizing, and explaining anomalies in cyber-physical systems. This is due to the ability of deep learning algorithms to learn complex patterns in data by using multiple layers of nonlinear processing in artificial neural networks. However, compared to traditional methods, deep learning has several advantages and disadvantages that need to be carefully considered when designing anomaly detection, localization, and explanation systems for cyber-physical systems.

One of the significant advantages of deep learning is its ability to detect complex patterns. Traditional methods may not be able to detect such complex patterns in data, especially in image and video data. Deep learning algorithms can detect anomalies that may not be visible to the naked eye or that may be difficult to identify using traditional image processing techniques. In addition, deep learning algorithms can achieve higher accuracy in detecting anomalies compared to traditional methods. Deep learning models can learn from large amounts of data and can generalize well to new data, whereas traditional methods may be limited by their reliance on handcrafted features.

Another advantage of deep learning in anomaly detection, localization, and explanation in cyber-physical systems is its ability to localize anomalies. Deep learning can identify the specific area of an image or video where an anomaly is occurring, which can be useful in determining the cause of the anomaly. This localization of anomalies helps to isolate the issue and minimize the impact on the system.

Moreover, deep learning can provide explanations for why an anomaly was detected in a cyber-physical system. By highlighting the features in an image or video that led to the detection of an anomaly, deep learning helps in understanding the root cause of the anomaly. This explanation can help in developing a better solution to the problem, and the system can be made more robust in the future [150].

Despite the advantages of deep learning, there are some limitations that must be considered. One such limitation is the need for large amounts of data to train deep learning algorithms effectively. This can be a challenge in cyber-physical systems where data may be scarce or difficult to collect. Furthermore, deep learning algorithms are computationally intensive and require powerful hardware to run efficiently, which can be a challenge in cyber-physical systems where resources may be limited [172].

Interpretability is another issue with deep learning. Deep learning models can be difficult to interpret, which can be a disadvantage in cyber-physical systems where it is important to understand the reasoning behind the detection of anomalies. Traditional methods may be more transparent and easier to interpret, and thus may be preferred in certain cases.

Finally, deep learning models may not be robust to changes in the environment or to adversarial attacks. This can be a concern in cyber-physical systems, where anomalies may be caused by unexpected changes in the environment or by intentional attacks. Thus, deep learning algorithms must be designed to be robust to such changes in the environment.

In conclusion, deep learning is a powerful tool for anomaly detection, localization, and explanation in cyber-physical systems. It has several advantages over traditional methods, including the ability to detect complex patterns, achieve higher accuracy, and localize anomalies. However, there are also limitations that must be considered, such as the need for large amounts of data, high computational costs, interpretability issues, and robustness concerns. By carefully considering these advantages and disadvantages, designers can develop effective anomaly detection, localization, and explanation systems for cyber-physical systems.

## 7.4 Challenge with Limited Data and Privacy Concerns

The scarcity of training data is one of the main obstacles to creating deep learning models for CPSs. The high cost and complexity of data collection and labeling, as well as safety and ethical concerns that restrict data collection in some domains, are frequently to blame for data scarcity. For instance, gathering data for autonomous driving may necessitate long driving distances in a variety of weather and traffic conditions, which can be costly and unfeasible. Similar risks or privacy issues may arise when gathering data for medical devices or essential infrastructure. Researchers may investigate methods like transfer learning, data augmentation, or semi-supervised learning, which can make use of existing data or produce synthetic data to expand the training set, to address the problem of data scarcity [178].

Data quality issues can significantly affect the performance and robustness of deep learning models, which heavily depend on the quality of the training data. Noise, outliers, missing values, consistency, bias, and imbalance are a few examples of data quality issues. For instance, noise or sensor failures in CPSs can cause sensor data to be inaccurate or misleading. Furthermore, bias in the data could be present as a result of the sampling strategy or the underlying data generation process . Models for CPSs must be robust and adaptable to different scenarios because they frequently operate in dynamic and uncertain environments. However, due to the high dimensionality of input spaces, the variability of output spaces, and the unpredictability of external factors, gathering diverse and representative data for CPSs can be difficult. For industrial robots, for instance, training data may need to include a variety of possible production-related tasks, objects, and disturbances.

CPSs frequently gather and handle delicate or private data, such as user behavior or health status, which raises privacy and security issues. Due to the distributed nature of data sources, the absence of clear ownership or consent mechanisms, and the potential for data leakage or misuse, it is difficult to ensure data privacy in CPSs. For instance, training data for wearable technology may need to safeguard user identities and health information from unauthorized access or disclosure. Researchers can use methods like differential privacy, federated learning to address the problem of data privacy. These methods can maintain the accuracy and usefulness of the models while protecting the privacy and confidentiality of the training data [132].

## 7.5 Causality vs Correlation

Two crucial ideas, causality and correlation, are frequently used in many branches of science and engineering, including cyber-physical systems. Anomalies can occur in these systems for a number of reasons, including hardware problems, software bugs, cyberattacks, and environmental disturbances. Understanding the causal connections between the various parts of the system and the correlations between the observed data are essential for localizing and explaining these anomalies.

The concept of causality describes the connection between a first event—the cause—and a second event—the effect—where the latter is a result of the former [20]. By tracing the sequence of events that led to an anomaly, causality in cyber-physical systems can be used to determine its primary cause. For instance, causality analysis can be used to determine the potential causes of an abnormal temperature reading from a sensor in a smart building system, such as a broken HVAC system, a defective sensor, or a cyberattack that changed the sensor data. Comparatively, correlation is the statistical association between two or more variables that frequently occur together. Since the observed relationship could be fictitious or coincidental, a correlation does not necessarily imply a causal connection. Correlation analysis can be used in cyber-physical systems to find patterns and trends in the data, which may reveal information about the system's underlying mechanisms. For instance, if a group of sensors in a manufacturing facility show a high degree of correlation in their readings, it may be an indication that they are measuring the same process variable and that a problem with one of the sensors will have an impact on the others [20].

Though both causality and correlation are crucial for locating and explaining anomalies in cyber-physical systems, they each have unique advantages and disadvantages. An in-depth knowledge of the dynamics of the system as well as the capacity to observe the pertinent events are necessary for causality analysis, which is useful for determining the cause of an anomaly. The complexity of the system or a lack of adequate data can frequently place constraints on causality analysis. On the other hand, correlation analysis is useful for spotting patterns and trends in the data but does not offer a reason for the observed relationships. Confounding variables or hidden factors that are not measured or observed may also have an impact on correlation analysis. To provide a thorough understanding of the system behavior, it is crucial to combine both causality and correlation analysis.

## 7.6 Balancing Accuracy and Interpretability

It is crucial for deep learning models to strike a balance between interpretability and accuracy. The balance between accuracy and interpretability is, as previously mentioned, greatly influenced by the use cases and the consequences of the model's decisions. In some situations, like self-driving cars and facial recognition, the main objective is to make accurate predictions, so interpretability may not be as important. However, in other applications, such as healthcare and finance, interpretability is crucial and accuracy alone is insufficient [109].

Because the model's choices could have a big impact on the patients, interpretability is important in the healthcare industry. To decide on diagnoses and treatments, medical professionals use the model as a guide. A model that is difficult to understand may lead

to decisions that are biased or incorrect, which could result in misdiagnoses or ineffective treatments. For instance, a model that correctly predicts the likelihood that a patient will have a specific disease but is unable to explain how it arrived at that conclusion is useless to medical professionals. To choose the best course of treatment, they must be aware of the characteristics that go into the diagnosis. Medical professionals can diagnose illnesses accurately and prescribe efficient treatments with the aid of an interpretable model, which can offer insights into the decision-making process.

Similar to this, interpretability in finance is important because poor choices can lead to sizable losses in money. Investors rely on models to help them decide which stocks to buy or sell, and an unintelligible model can lead to poor investment choices. For instance, a model that correctly forecasts the performance of a specific stock but is unable to articulate how it arrived at that conclusion is of no use to investors. To make wise decisions about buying or selling the stock, they must be aware of the variables that affect its performance. An interpretable model can offer insights into the decision-making process and assist investors in making wise investment decisions [29]. An important consideration in balancing accuracy and interpretability in deep learning models is the trade-off between model complexity and model performance. Deep neural networks, for example, are complex models that learn intricate representations from a large amount of data to produce high accuracy. The decision-making process can be difficult to understand because these models are frequently difficult to understand. In applications where the consequences of poor decisions can be severe, the lack of interpretability can be a significant drawback.

Simpler models, on the other hand, may be less accurate but are interpretable, such as decision trees and linear models. These models are frequently employed in fields like healthcare and finance where the ability to interpret results is crucial. The most important features that go into making a decision can be found using decision trees. These models might not, however, be strong enough to learn intricate representations from massive amounts of data. Understanding the application and the effects of the model's decisions in great detail is necessary to strike the right balance between accuracy and interpretability. Accuracy is crucial in some applications, like self-driving cars, while interpretability may not be required. In contrast, accuracy alone is insufficient in the fields of finance and healthcare, where interpretability is essential. Utilizing hybrid models,like the method proposed in this thesis, combine the benefits of deep learning models and shallow models, is one method for striking a balance between accuracy and interpretability. Hybrid models can make precise predictions while also illuminating the decision-making process. A neural network, for instance, can learn intricate representations from data, and a decision tree can clarify the model's predictions by determining the most important features. A deep neural network and a decision tree can be used to make precise predictions and gain understanding of the decision-making process.

Utilizing model-agnostic methods like LIME (Local Interpretable Model-Agnostic Explanations), SHAP (Shapley Additive Explanations) [56], and CAM (Class Activation Maps)[185] is another strategy for striking a balance between accuracy and interpretability. By locating the most pertinent features that contribute to a specific decision, these techniques can offer insights into the decision-making process of complex models, such as deep neural networks. These methods can be used with any model, making them model-neutral, and they can give understandable justifications for complicated models.

In conclusion, striking a balance between model complexity and model performance is

essential for deep learning models in order to maintain accuracy and interpretability. While simpler models like decision trees are understandable but may not be as accurate as more complex models like deep neural networks, they can achieve high accuracy at the expense of frequently being difficult to interpret. By balancing precision and interpretability, hybrid models and model-agnostic methods can make precise predictions while also illuminating the decision-making process. Understanding the application and the effects of the model's decisions in great detail is necessary to strike the right balance between accuracy and interpretability.

## 7.7 Importance of Interpretability and Explainability

There is a large emphasis placed on interpretability and explainability in deep learning models because the effects of errors in CPSs can be severe and widespread. An error in classifying a pedestrian or an obstruction, for instance, could cause a fatal accident in an autonomous vehicle. In the same way, patients' lives could be in danger in healthcare systems if a misdiagnosis or bad treatment advice is given. As a result, the capability to comprehend and validate the deep learning models' decision-making process can aid in ensuring the security and dependability of CPSs.

Additionally, the system's interpretability and explainability can aid in building confidence and trust, which can increase usage and adoption. Users may be hesitant to trust predictions or recommendations made by a deep learning model if they do not understand how it operates. On the other hand, users may be more inclined to trust the system and use it more frequently if they can comprehend and validate the decision-making process.

Finally, by pointing out potential areas for improvement, interpretability and explainability can aid in enhancing the performance of deep learning models. For instance, if a model consistently misclassifies certain input types, an understandable explanation can assist in locating the feature or features responsible. The model can then be improved upon using this data in order to increase accuracy.

There exist a varity of methods that can aid in both explanablity and interpretability. Some popular and promising approaches are discussed here.

## 7.8 Potential of Causal Models

### 7.8.1 Promising Techniques

There exist a varity of methods that can aid in both explanablity and interpretability. Some popular and promising approaches are discussed here.

Saliency maps display the input features that have the greatest impact on the predictions made by the model. By computing the output's gradient with respect to the input features and projecting the gradients back onto the input space, these maps can be made. The input features that have the greatest impact on the output are highlighted on the final map. The decision-making process of the model can be explained intuitively and visually with saliency maps, which can aid users in understanding how the model arrived at its prediction [160].

Local surrogate models simulate the deep learning model's behavior in a particular area of the input space. These models may be easier to understand and more straightforward

than the original deep learning model, and they can aid in finding the pertinent features for a given prediction. The LIME (Local Interpretable Model-Agnostic Explanations) method produces an interpretable model that roughly approximates the deep learning model's behavior for a given input. Local surrogate models can be trained using LIME. The resulting model can assist in explaining how the deep learning model makes decisions in the local area of the input space [197].

Counterfactual explanations entail altering the input characteristics to show how doing so would result in different results. The critical characteristics that influence the model's predictions can be found using this technique, which can also reveal how the model responds to various scenarios. One method for creating counterfactual explanations is to use a generative model, like a variational autoencoder (VAE), to create alternative input samples that are similar to the original input but result in different predictions. Another strategy is to modify the input features using an optimization algorithm in a way that complies with certain requirements, such as keeping the input's similarity to the original input while producing a different prediction [79].

These techniques can aid in improving the interpretability of deep learning models in CPSs by revealing the model's decision-making process, locating the necessary input features for prediction, and describing how the model behaves in various scenarios. The causal connections between the input features and the output predictions are not always captured by these methods, though. By using directed acyclic graphs (DAGs) to encode the conditional interdependencies between variables, causality models offer a more thorough means of comprehending the behavior of deep learning models.

### 7.8.2 Advantanges and Disadvantages of Causal Models

The goal of causality models, also referred to as causal inference models, is to identify the relationships between variables in a system that cause and result in certain outcomes. For the purpose of encoding conditional interdependencies between variables and facilitating analysis of the effects of interventions on the system, these models employ directed acyclic graphs (DAGs). By identifying the features that directly or indirectly influence the output and how changing these features would affect the prediction, causality models can be used in the context of deep learning to explain how the model makes its predictions.

The ability to provide a more thorough and transparent understanding of deep learning models is one of the benefits of causality models. The effectiveness of deep learning models can be verified and validated, and potential sources of bias or error can be found, by revealing the causal relationships between input features and output predictions. The behavior of a model can also be interpreted using causality models in terms of concepts that are more easily understood by humans, such as how the weather affects autonomous vehicles or how patient characteristics affect medical diagnoses [174].

However, there are issues with causality models when it comes to CPSs. Data scarcity is one of the main problems. The diversity and complexity of the system must be captured in a substantial amount of data before a causality model can be developed. However, gathering and labeling such data for CPSs can be expensive, time-consuming, and occasionally impractical due to physical limitations or ethical considerations.

The complexity of the model is another difficulty. As the number of variables and interactions rises, causality models can easily become complicated. This complexity raises

the risk of overfitting, causes significant computational expense, and makes it challenging to understand the model's output. Furthermore, causality models might need prior knowledge of the system and domain expertise that isn't always accessible or available [54].

Another difficulty that causality models face is domain adaptation. Adaptability and flexibility are essential for CPSs because they frequently operate in dynamic and unpredictable environments. As a result, causality models must be able to handle system changes like the addition of new variables or contexts while still remaining accurate and understandable. Though it may necessitate more data or model retraining, which can be time- and resource-intensive, adapting causality models to new domains may be necessary.

Further study is required to create causality models for deep learning interpretability in CPSs that are effective and efficient to overcome these issues. Designing effective and scalable causality algorithms that can manage complex and high-dimensional systems, investigating the use of transfer learning and domain adaptation techniques to improve the model's generalization ability, and developing novel data collection and labeling techniques that are tailored to CPSs are a few examples of the research that could be done in this area. Research on the combination of causality models and other interpretability techniques may also aid in combining their advantages and overcoming their drawbacks.

## 7.9 Future Work

Anomaly detection is an essential task in ensuring the security and reliability of cyber physical systems. The proposed model, which uses a deep learning-based LSTM encoder-decoder model with PCA localisation and LSTM reconstruction error per feature, has shown promising results in detecting anomalies in a specific cyber physical system. However, to evaluate the model's generalisation ability and identify potential challenges that may arise in different settings, future works should extend the research to other cyber physical systems.

Testing the proposed model on different cyber physical systems can help determine its effectiveness in detecting anomalies in various domains. It can also identify potential challenges that may arise in different settings, such as variations in data types, data sources, and system architectures. By testing the model on various cyber physical systems, researchers can evaluate its adaptability and generalisation ability.

One possible approach to test the proposed model on different cyber physical systems is to use publicly available datasets. Many datasets exist that represent different cyber physical systems, such as power grids, transportation systems, and medical devices. Researchers can use these datasets to evaluate the model's performance and identify any weaknesses or areas of improvement.

Another approach to testing the proposed model on different cyber physical systems is to collaborate with industry partners. Industry partners can provide access to real-world data from their systems, which can help researchers evaluate the model's effectiveness in detecting anomalies in practical scenarios. Such collaborations can also help identify any practical challenges that may arise in deploying the model in a production environment.

In addition to testing the proposed model on different cyber physical systems, future works can also explore the possibility of extending the model to work with multiple data

modalities. Many cyber physical systems generate different types of data, such as sensor data, log data, and network data. By developing a multi-modal approach, researchers can integrate different types of data into the anomaly detection process, which can help improve the accuracy and effectiveness of the model.

### 7.9.1 Focus on Explainability

The development of explainable artificial intelligence (XAI)[109] has become increasingly important in recent years, particularly in applications such as anomaly detection in cyber physical systems. Although deep learning models have shown remarkable performance in detecting anomalies, their black-box nature often makes it challenging to understand how and why a particular decision was made. Therefore, enhancing the explainability of the model has become a crucial aspect of anomaly detection in cyber physical systems.

The proposed model employs a supervised tree to generate if-then rules and a causal Bayesian network to explain anomalies. While these methods are effective in providing some level of interpretability, future works can explore other explainability methods, particularly causal methods, to enhance the model's explainability [29].

Causal methods are an approach to XAI that aims to identify the causal relationships between the input features and the output variable. By understanding the causal relationships, it becomes possible to explain why a particular decision was made by the model. There are several causal methods that can be explored to enhance the explainability of the model, including counterfactual explanations, feature importance, and saliency maps [15].

### 7.9.2 Exploring Other Detection Techniques

Anomaly detection using deep learning in cyber physical systems is a complex problem that requires careful consideration of the model architecture and hyperparameters. The proposed model in this thesis employs an LSTM encoder-decoder and PCA localisation or LSTM reconstruction error per feature to detect anomalies in the system. However, to further improve the model's performance, future works can explore different deep learning architectures and adjust the hyperparameters of the model.

One possible approach is to explore different recurrent neural network (RNN) architectures. The LSTM encoder-decoder architecture used in the proposed model is one type of RNN, but other types, such as gated recurrent unit (GRU)[43] or bidirectional LSTM [154], can be explored to improve the model's accuracy. These architectures can be combined with other techniques, such as attention mechanisms, to further enhance the model's performance.

Another approach is to adjust the hyperparameters of the model. Hyperparameters, such as learning rate, batch size, and number of epochs, can significantly impact the model's performance. Fine-tuning these hyperparameters using techniques such as grid search or Bayesian optimisation can help improve the model's accuracy and generalisation ability.

# Chapter 8

# Conclusion

In conclusion, the Seq-2-Seq LSTM Encoder Decoder has demonstrated promising results on both the Tennessee Eastman simulated data and benchmark datasets. This technique has shown to be effective in detecting and localizing anomalies in cyber physical systems, which is a crucial task in ensuring the safety and reliability of these systems. The localisation technique used in this study has also proven to be a viable baseline technique.

Furthermore, the causal network and GBM model have both been effective in providing explanations for anomalies, with each having their own advantages and disadvantages. The causal network provides more interpretability and transparency, while the GBM model has higher predictive accuracy. Therefore, depending on the specific needs and goals of the application, either method could be chosen for explanation.

The framework presented in this study serves as a proof of concept and a rudimentary framework that can be built upon and improved for anomaly detection in cyber physical systems. Although the main goal of presenting a prototype for a unified framework to detect, localize, and explain anomalies has been achieved, there is still a lot of room for exploring improvements to the framework.

Overall, the results of this study demonstrate the effectiveness of the Seq-2-Seq LSTM Encoder Decoder in detecting and localizing anomalies in cyber physical systems, as well as the potential of the causal network and GBM model for providing explanations. The framework presented in this study can serve as a starting point for future research in this area, and can potentially lead to the development of more advanced and sophisticated techniques for anomaly detection and explanation in cyber physical systems.

# Appendix A

# Appendix

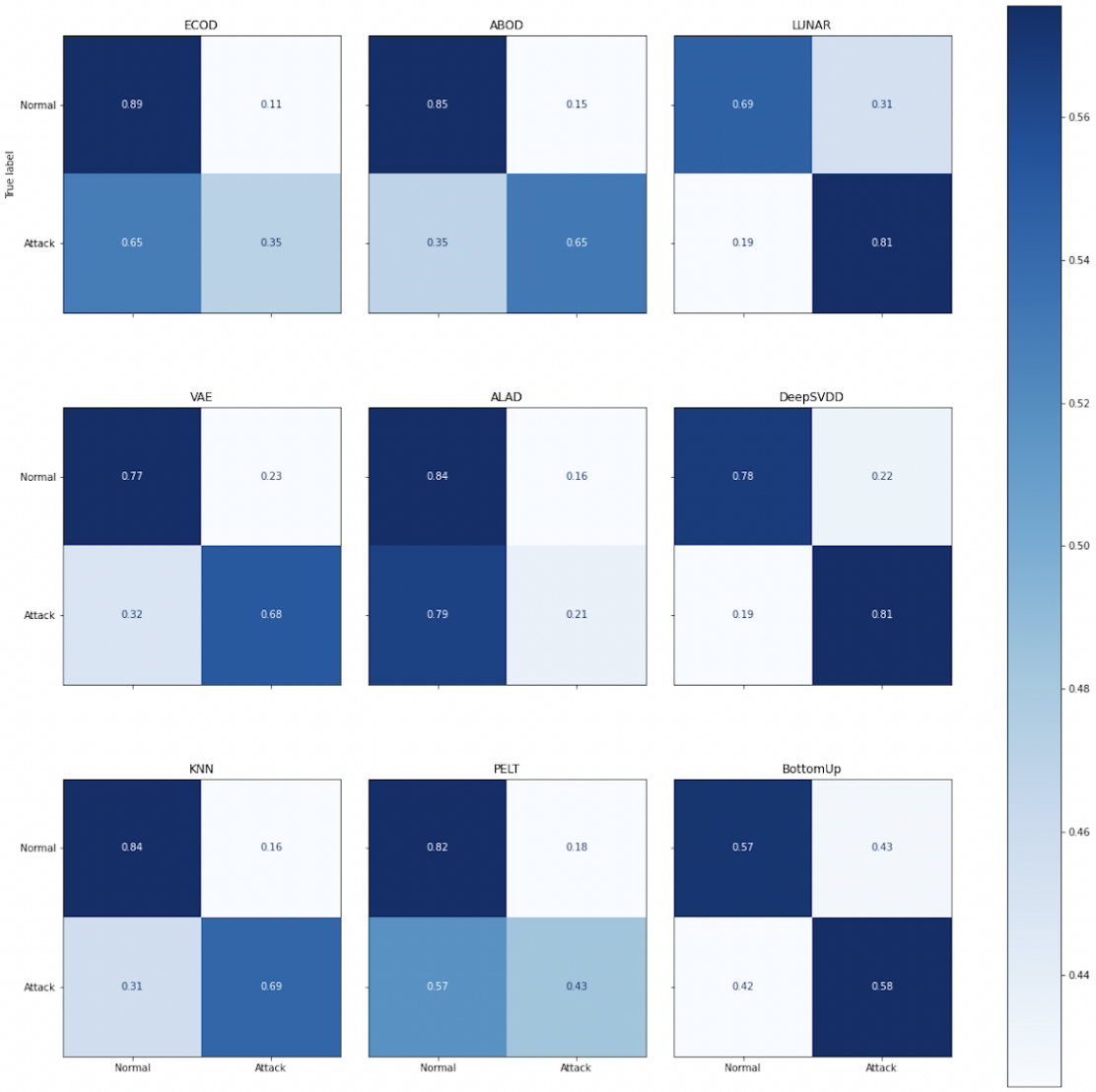



Figure A.1: Proposed Model Confusion Matrix - Single Fault

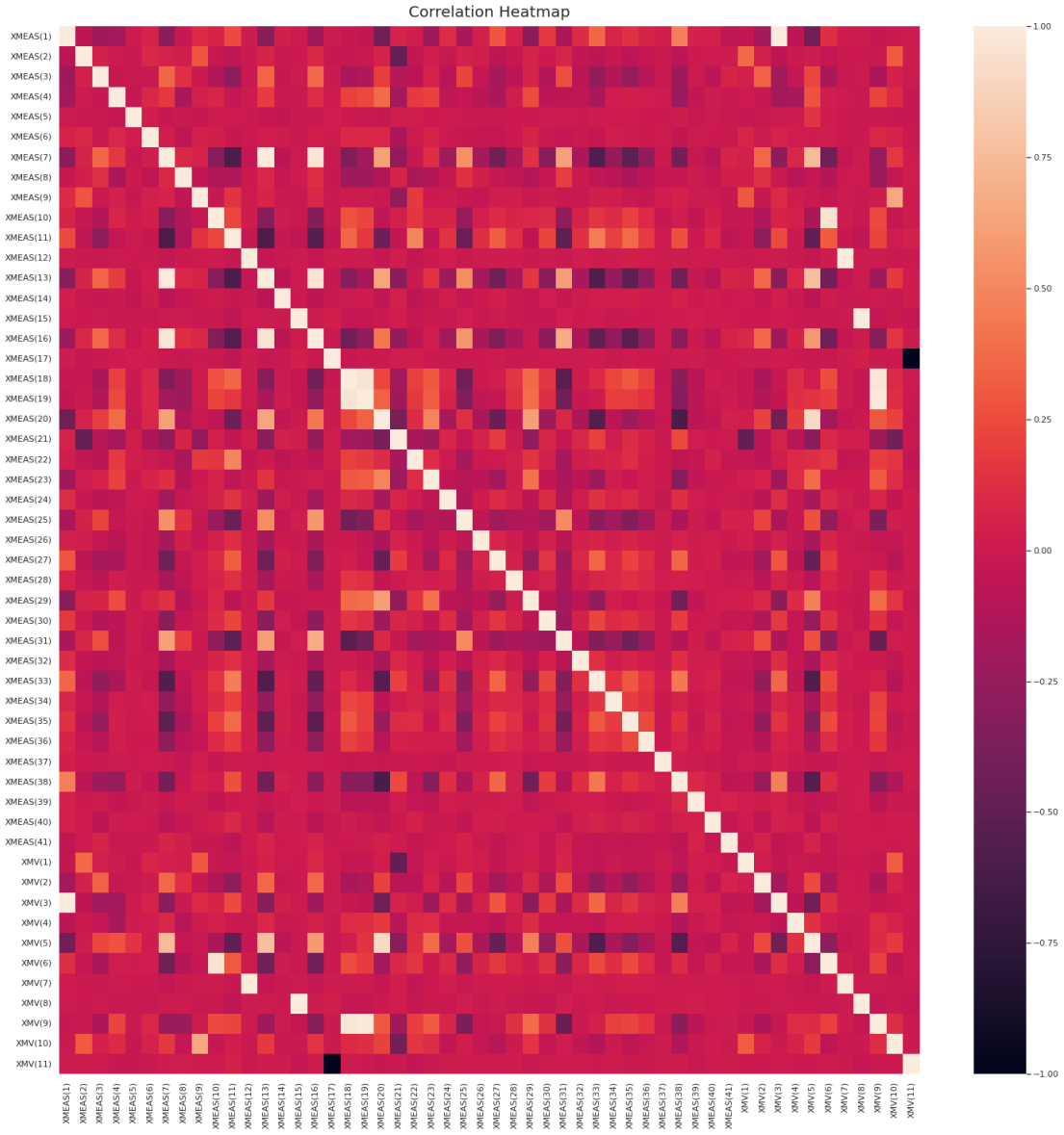



Figure A.2: Spearman Rank Correlation Coefficient for TEP

| Subsection Name | Component Name | Feature Name |
|---|---|---|
| Input feed | A feed (stream 1) | XMEAS(1) |
| | D feed (stream 2) | XMEAS(2) |
| | E feed (stream 3) | XMEAS(3) |
| | A and C feed | XMEAS(4) |
| Reactor | Reactor feed rate | XMEAS(6) |
| | Reactor pressure | XMEAS(7) |
| | Reactor level | XMEAS(8) |
| | Reactor temperature | XMEAS(9) |
| Separator | Separator temperature | XMEAS(11) |
| | Separator level | XMEAS(12) |
| | Separator pressure | XMEAS(13) |
| | Separator underflow | XMEAS(14) |
| Stripper | Stripper level | XMEAS(15) |
| | Stripper pressure | XMEAS(16) |
| | Stripper underflow | XMEAS(17) |
| | Stripper temperature | XMEAS(18) |
| | Stripper steam flow | XMEAS(19) |
| Miscellaneous | Recycle flow | XMEAS(5) |
| | Purge rate | XMEAS(10) |
| | Compressor work | XMEAS(20) |
| | Reactor water temperature | XMEAS(21) |
| | Separator water temperature | XMEAS(22) |
| Reactor feed analysis | Component A | XMEAS(23) |
| | Component B | XMEAS(24) |
| | Component C | XMEAS(25) |
| | Component D | XMEAS(26) |
| | Component E | XMEAS(27) |
| | Component F | XMEAS(28) |
| Purge gas analysis | Component A | XMEAS(29) |
| | Component B | XMEAS(30) |
| | Component C | XMEAS(31) |
| | Component D | XMEAS(32) |
| | Component E | XMEAS(33) |
| | Component F | XMEAS(34) |
| | Component G | XMEAS(35) |
| | Component H | XMEAS(36) |
| Product analysis | Component D | XMEAS(37) |
| | Component E | XMEAS(38) |
| | Component F | XMEAS(39) |
| | Component G | XMEAS(40) |
| | Component H | XMEAS(41) |

Figure A.3: Tennessee Eastman - Process Variables

# Bibliography


[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. "On the surprising behavior of distance metrics in high dimensional space". In: *International conference on database theory*. Springer. 2001, pp. 420–434.

[2] Kazi Masudul Alam and Abdulmotaleb El Saddik. "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems". In: *IEEE access* 5 (2017), pp. 2050–2062.

[3] David M Allen. "Mean square error of prediction as a criterion for selecting variables". In: *Technometrics* 13.3 (1971), pp. 469–475.

[4] Ahmed Abdulhasan Alwan et al. "Data quality challenges in large-scale cyber-physical systems: A systematic review". In: *Information Systems* 105 (2022), p. 101951.

[5] Tsatsral Amarbayasgalan et al. "Unsupervised anomaly detection approach for time-series in multi-domains using deep reconstruction error". In: *Symmetry* 12.8 (2020), p. 1251.

[6] Tsatsral Amarbayasgalan et al. "Unsupervised anomaly detection approach for time-series in multi-domains using deep reconstruction error". In: *Symmetry* 12.8 (2020), p. 1251.

[7] Redhwan Al-amri et al. "A review of machine learning and deep learning techniques for anomaly detection in IoT data". In: *Applied Sciences* 11.12 (2021), p. 5320.

[8] Jinwon An and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability". In: *Special Lecture on IE* 2.1 (2015), pp. 1–18.

[9] Agustin Garcia Asuero, Ana Sayago, and AG González. "The correlation coefficient: An overview". In: *Critical reviews in analytical chemistry* 36.1 (2006), pp. 41–59.

[10] Ivan E Auger and Charles E Lawrence. "Algorithms for the optimal identification of segment neighborhoods". In: *Bulletin of mathematical biology* 51.1 (1989), pp. 39–54.

[11] Radhakisan Baheti and Helen Gill. "Cyber-physical systems". In: *The impact of control technology* 12.1 (2011), pp. 161–166.

[12] Dor Bank, Noam Koenigstein, and Raja Giryes. "Autoencoders". In: *arXiv preprint arXiv:2003.05991* (2020).

[13] Piero Baraldi et al. "Comparison of data-driven reconstruction methods for fault detection". In: *IEEE Transactions on Reliability* 64.3 (2015), pp. 852–860.

[14] Andreas Bathelt, N Lawrence Ricker, and Mohieddine Jelali. "Revision of the Tennessee Eastman process model". In: *IFAC-PapersOnLine* 48.8 (2015), pp. 309–314.

[15] Sander Beckers. "Causal explanations and XAI". In: *Conference on Causal Learning and Reasoning*. PMLR. 2022, pp. 90–109.

[16] James Bergstra, Daniel Yamins, and David Cox. "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: *International conference on machine learning*. PMLR. 2013, pp. 115–123.

[17] James Bergstra et al. "Algorithms for hyper-parameter optimization". In: *Advances in neural information processing systems* 24 (2011).

[18] Anatolij Bezemskij et al. "Detecting cyber-physical threats in an autonomous robotic vehicle using Bayesian networks". In: *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2017, pp. 98–103.

[19] Jock A Blackard and Denis J Dean. "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables". In: *Computers and electronics in agriculture* 24.3 (1999), pp. 131–151.

[20] Hubert M Blalock Jr. "Correlation and causality: The multivariate case". In: *Social Forces* 39.3 (1961), pp. 246–251.

[21] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[22] Francesco Bodria et al. "Benchmarking and survey of explanation methods for black box models". In: *arXiv preprint arXiv:2102.13076* (2021).

[23] Markus M Breunig et al. "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[24] Christopher P Burgess et al. "Understanding disentangling in beta-VAE". In: *arXiv preprint arXiv:1804.03599* (2018).

[25] Matthew Burruss, Shreyas Ramakrishna, and Abhishek Dubey. "Deep-rbf networks for anomaly detection in automotive cyber-physical systems". In: *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE. 2021, pp. 55–60.

[26] Feiyang Cai and Xenofon Koutsoukos. "Real-time out-of-distribution detection in learning-enabled cyber-physical systems". In: *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE. 2020, pp. 174–183.

[27] José Camacho et al. "PCA-based multivariate statistical network monitoring for anomaly detection". In: *Computers & Security* 59 (2016), pp. 118–137.

[28] Francesca Capaci et al. "The revised Tennessee Eastman process simulator as testbed for SPC and DoE methods". In: *Quality Engineering* 31.2 (2019), pp. 212–229.

[29] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. "Machine learning interpretability: A survey on methods and metrics". In: *Electronics* 8.8 (2019), p. 832.

[30] Gavneet Singh Chadha et al. "Deep Convolutional Clustering-Based Time Series Anomaly Detection". In: *Sensors* 21.16 (2021), p. 5488.

[31] Kaustav Chatterjee, V Padmini, and SA Khaparde. "Review of cyber attacks on power system operations". In: *2017 IEEE Region 10 Symposium (TENSYMP)*. IEEE. 2017, pp. 1–6.

[32] Honghua Chen et al. "Fault Diagnosis of the Dynamic Chemical Process Based on the Optimized CNN-LSTM Network". In: *ACS omega* 7.38 (2022), pp. 34389–34400.

[33] Jinghui Chen et al. "Outlier detection with autoencoder ensembles". In: *Proceedings of the 2017 SIAM international conference on data mining*. SIAM. 2017, pp. 90–98.

[34] Zhaomin Chen et al. "Autoencoder-based network anomaly detection". In: *2018 Wireless telecommunications symposium (WTS)*. IEEE. 2018, pp. 1–5.

[35] Jui-Sheng Chou and Abdi Suryadinata Telaga. "Real-time detection of anomalous power consumption". In: *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 400–411.

[36] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. "Anomaly detection for IoT time-series data: A survey". In: *IEEE Internet of Things Journal* 7.7 (2019), pp. 6481–6494.

[37] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[38] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[39] Claire D'Este, Michael Towsey, and Joachim Diederich. "Sparsely-connected recurrent neural networks for natural language learning". In: *First Workshop on Natural Language Processing and Neural Networks. Beijing, China*. 1999, pp. 64–69.

[40] Lucas Deecke et al. "Transfer-based semantic anomaly detection". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2546–2558.

[41] Wenfeng Deng et al. "LSTMED: An uneven dynamic process monitoring method based on LSTM and Autoencoder neural network". In: *Neural Networks* (2022).

[42] Patricia Derler, Edward A Lee, and Alberto Sangiovanni Vincentelli. "Modeling cyber–physical systems". In: *Proceedings of the IEEE* 100.1 (2011), pp. 13–28.

[43] Rahul Dey and Fathi M Salem. "Gate-variants of gated recurrent unit (GRU) neural networks". In: *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE. 2017, pp. 1597–1600.

[44] Nan Ding et al. "Multivariate-time-series-driven real-time anomaly detection based on bayesian network". In: *Sensors* 18.10 (2018), p. 3367.

[45] Rui Ding et al. "BiGAN: collaborative filtering with bidirectional generative adversarial networks". In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM. 2020, pp. 82–90.

[46] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. "Adversarial feature learning". In: *arXiv preprint arXiv:1605.09782* (2016).

[47] A Rogier T Donders et al. "A gentle introduction to imputation of missing values". In: *Journal of clinical epidemiology* 59.10 (2006), pp. 1087–1091.

[48] James J Downs and Ernest F Vogel. "A plant-wide industrial process control problem". In: *Computers & chemical engineering* 17.3 (1993), pp. 245–255.

[49] Vincent Dumoulin et al. "Adversarially learned inference". In: *arXiv preprint arXiv:1606.00704* (2016).

[50] Dara Entekhabi et al. "The soil moisture active passive (SMAP) mission". In: *Proceedings of the IEEE* 98.5 (2010), pp. 704–716.

[51] Jerome Fan, Suneel Upadhye, and Andrew Worster. "Understanding receiver operating characteristic (ROC) curves". In: *Canadian Journal of Emergency Medicine* 8.1 (2006), pp. 19–20.

[52] Longji Feng et al. "Anomaly detection for electricity consumption in cloud computing: framework, methods, applications, and challenges". In: *EURASIP Journal on Wireless Communications and Networking* 2020.1 (2020), pp. 1–12.

[53] Pavel Filonov, Fedor Kitashov, and Andrey Lavrentyev. "Rnn-based early cyberattack detection for the tennessee eastman process". In: *arXiv preprint arXiv:1709.02232* (2017).

[54] Nir Friedman and Zohar Yakhini. "On the sample complexity of learning Bayesian networks". In: *arXiv preprint arXiv:1302.3579* (2013).

[55] Piotr Fryzlewicz. "Unbalanced Haar technique for nonparametric function estimation". In: *Journal of the American Statistical Association* 102.480 (2007), pp. 1318–1327.

[56] Marıa Vega Garcıa and José L Aznarte. "Shapley additive explanations for NO2 forecasting". In: *Ecological Informatics* 56 (2020), p. 101039.

[57] Alexander Geiger et al. "TadGAN: Time series anomaly detection using generative adversarial networks". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 33–43.

[58] Jonathan Goh et al. "Anomaly detection in cyber physical systems using recurrent neural networks". In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE. 2017, pp. 140–145.

[59] Adam Goodge et al. "Lunar: Unifying local outlier detection methods via graph neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 6. 2022, pp. 6737–6745.

[60] John P Grotzinger et al. "Mars Science Laboratory mission and science investigation". In: *Space science reviews* 170.1 (2012), pp. 5–56.

[61] Jiuxiang Gu et al. “Recent advances in convolutional neural networks”. In: *Pattern recognition* 77 (2018), pp. 354–377.

[62] Ishaan Gulrajani et al. “Improved training of wasserstein gans”. In: *Advances in neural information processing systems* 30 (2017).

[63] Salem Hadim and Nader Mohamed. “Middleware: Middleware challenges and approaches for wireless sensor networks”. In: *IEEE distributed systems online* 7.3 (2006), pp. 1–1.

[64] Nastaran Hajarian, Farzad Movahedi Sobhani, and Seyed Jafar Sadjadi. “An improved approach for fault detection by simultaneous overcoming of high-dimensionality, autocorrelation, and time-variability”. In: *Plos one* 15.12 (2020), e0243146.

[65] Songqiao Han et al. “Adbench: Anomaly detection benchmark”. In: *arXiv preprint arXiv:2206.09426* (2022).

[66] Songqiao Han et al. “Adbench: Anomaly detection benchmark”. In: *arXiv preprint arXiv:2206.09426* (2022).

[67] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[68] Michael A Hayes and Miriam AM Capretz. “Contextual anomaly detection framework for big sensor data”. In: *Journal of Big Data* 2.1 (2015), pp. 1–22.

[69] Jiun Tian Hoe et al. “One loss for all: Deep hashing with a single cosine similarity based learning objective”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24286–24298.

[70] Mohammad Hossin and Md Nasir Sulaiman. “A review on evaluation metrics for data classification evaluations”. In: *International journal of data mining & knowledge management process* 5.2 (2015), p. 1.

[71] Jin Huang and Charles X Ling. “Using AUC and accuracy in evaluating learning algorithms”. In: *IEEE Transactions on knowledge and Data Engineering* 17.3 (2005), pp. 299–310.

[72] Kyle Hundman et al. “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 387–395.

[73] Mikhail Hushchyn. “Change Point Detection”. In: ().

[74] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. “Sequential model-based optimization for general algorithm configuration”. In: *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*. Springer. 2011, pp. 507–523.

[75] Md Rakibul Islam et al. “GLAD: GLocalized Anomaly Detection via Human-in-the-Loop Learning”. In: *arXiv preprint arXiv:1810.01403* (2018).

[76] Brad Jackson et al. “An algorithm for optimal partitioning of data on an interval”. In: *IEEE Signal Processing Letters* 12.2 (2005), pp. 105–108.

[77] Lin Jiang et al. “Anomaly detection of industrial multi-sensor signals based on enhanced spatiotemporal features”. In: *Neural Computing and Applications* (2022), pp. 1–13.

[78] Tao Jiang et al. “Discriminative reconstruction constrained generative adversarial network for hyperspectral anomaly detection”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.7 (2020), pp. 4666–4679.

[79] Fredrik Johansson, Uri Shalit, and David Sontag. “Learning representations for counterfactual inference”. In: *International conference on machine learning*. PMLR. 2016, pp. 3020–3029.

[80] James M Joyce. “Kullback-leibler divergence”. In: *International encyclopedia of statistical science*. Springer, 2011, pp. 720–722.

[81] Maxim O Kalinin, Daria S Lavrova, and AV Yarmak. “Detection of threats in cyberphysical systems based on deep learning methods using multidimensional time series”. In: *Automatic control and computer sciences* 52.8 (2018), pp. 912–917.

[82] Stamatis Karnouskos. “Stuxnet worm impact on industrial cyber-physical system security”. In: *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2011, pp. 4490–4494.

[83] Eamonn Keogh et al. “An online algorithm for segmenting time series”. In: *Proceedings 2001 IEEE international conference on data mining*. IEEE. 2001, pp. 289–296.

[84] Tung Kieu et al. “Outlier Detection for Time Series with Recurrent Autoencoder Ensembles.” In: *IJCAI*. 2019, pp. 2725–2732.

[85] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. “Optimal detection of changepoints with a linear computational cost”. In: *Journal of the American Statistical Association* 107.500 (2012), pp. 1590–1598.

[86] Taehyeon Kim et al. “Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation”. In: *arXiv preprint arXiv:2105.08919* (2021).

[87] Masanari Kimura and Takashi Yanagihara. “Anomaly detection using GANs for visual inspection in noisy training data”. In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 373–385.

[88] Timo Klerx et al. “Model-based anomaly detection for discrete event systems”. In: *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE. 2014, pp. 665–672.

[89] William Knowles et al. “A survey of cyber security management in industrial control systems”. In: *International journal of critical infrastructure protection* 9 (2015), pp. 52–80.

[90] Teuvo Kohonen. “The self-organizing map”. In: *Proceedings of the IEEE* 78.9 (1990), pp. 1464–1480.

[91] Andrei V Konstantinov and Lev V Utkin. “Interpretable machine learning with an ensemble of gradient boosting machines”. In: *Knowledge-Based Systems* 222 (2021), p. 106993.

[92] Ana Kovacevic and Dragana Nikolic. "Cyber attacks on critical infrastructure: Review and challenges". In: *Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance* (2015), pp. 1–18.

[93] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. "Angle-based outlier detection in high-dimensional data". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 444–452.

[94] Sudha Krishnamurthy, Soumik Sarkar, and Ashutosh Tewari. "Scalable anomaly detection and isolation in cyber-physical systems using bayesian networks". In: *Dynamic Systems and Control Conference*. Vol. 46193. American Society of Mechanical Engineers. 2014, V002T26A006.

[95] Sudha Krishnamurthy, Soumik Sarkar, and Ashutosh Tewari. "Scalable anomaly detection and isolation in cyber-physical systems using bayesian networks". In: *Dynamic Systems and Control Conference*. Vol. 46193. American Society of Mechanical Engineers. 2014, V002T26A006.

[96] Wenfu Ku, Robert H Storer, and Christos Georgakis. "Disturbance detection and isolation by dynamic principal component analysis". In: *Chemometrics and intelligent laboratory systems* 30.1 (1995), pp. 179–196.

[97] Johan Kwisthout. "Most probable explanations in Bayesian networks: Complexity and tractability". In: *International Journal of Approximate Reasoning* 52.9 (2011), pp. 1452–1469.

[98] Alexander Lavin and Subutai Ahmad. "Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark". In: *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE. 2015, pp. 38–44.

[99] Alexander Lavin and Subutai Ahmad. "Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark". In: *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE. 2015, pp. 38–44.

[100] Aleksandar Lazarevic and Vipin Kumar. "Feature bagging for outlier detection". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 157–166.

[101] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[102] Edward Ashford Lee and Sanjit Arunkumar Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. Mit Press, 2016.

[103] Jay Lee, Behrad Bagheri, and Hung-An Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems". In: *Manufacturing letters* 3 (2015), pp. 18–23.

[104] Dan Li et al. "Anomaly detection with generative adversarial networks for multivariate time series". In: *arXiv preprint arXiv:1809.04758* (2018).

[105] Kun-Lun Li et al. "Improving one-class SVM for anomaly detection". In: *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*. Vol. 5. IEEE. 2003, pp. 3077–3081.

[106] Zheng Li et al. “Ecod: Unsupervised outlier detection using empirical cumulative distribution functions”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[107] Wenlong Liao et al. “A review of graph neural networks and their applications in power systems”. In: *Journal of Modern Power Systems and Clean Energy* (2021).

[108] Qin Lin et al. “TABOR: A graphical model-based approach for anomaly detection in industrial control systems”. In: *Proceedings of the 2018 on asia conference on computer and communications security*. 2018, pp. 525–536.

[109] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. “Explainable ai: A review of machine learning interpretability methods”. In: *Entropy* 23.1 (2020), p. 18.

[110] Zachary Chase Lipton, Charles Elkan, and Balakrishnan Narayanaswamy. “Thresholding classifiers to maximize F1 score”. In: *arXiv preprint arXiv:1402.1892* (2014).

[111] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[112] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).

[113] Yuan Luo et al. “Deep Learning-Based Anomaly Detection in Cyber-Physical Systems: Progress and Opportunities”. In: *ACM Comput. Surv.* 54.5 (May 2021). ISSN: 0360-0300. DOI: 10.1145/3453155. URL: https://doi.org/10.1145/3453155.

[114] Yuan Luo et al. “Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities”. In: *ACM Computing Surveys (CSUR)* 54.5 (2021), pp. 1–36.

[115] Andrei Manolache, Florin Brad, and Elena Burceanu. “Date: Detecting anomalies in text via self-supervision of transformers”. In: *arXiv preprint arXiv:2104.05591* (2021).

[116] Andre Martins and Ramon Astudillo. “From softmax to sparsemax: A sparse model of attention and multi-label classification”. In: *International conference on machine learning*. PMLR. 2016, pp. 1614–1623.

[117] Aditya P Mathur and Nils Ole Tippenhauer. “SWaT: A water treatment testbed for research and training on ICS security”. In: *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE. 2016, pp. 31–36.

[118] Stephen McLaughlin et al. “The cybersecurity landscape in industrial control systems”. In: *Proceedings of the IEEE* 104.5 (2016), pp. 1039–1057.

[119] Giovanni Menegozzo, Diego Dall’Alba, and Paolo Fiorini. “CIPCaD-Bench: Continuous Industrial Process datasets for benchmarking Causal Discovery methods”. In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2022, pp. 2124–2131.

[120] Yisroel Mirsky et al. “Kitsune: an ensemble of autoencoders for online network intrusion detection”. In: *arXiv preprint arXiv:1802.09089* (2018).

[121] Hossein Mohammadi Rouzbahani et al. "Anomaly detection in cyber-physical systems using machine learning". In: *Handbook of big data privacy*. Springer, 2020, pp. 219–235.

[122] Janice M Morse. *"Cherry picking": Writing from thin data*. 2010.

[123] Mohsin Munir et al. "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series". In: *Ieee Access* 7 (2018), pp. 1991–2005.

[124] Benson Mwangi, Jair C Soares, and Khader M Hasan. "Visualization and unsupervised predictive clustering of high-dimensional multimodal neuroimaging data". In: *Journal of neuroscience methods* 236 (2014), pp. 19–25.

[125] Alexey Natekin and Alois Knoll. "Gradient boosting machines, a tutorial". In: *Frontiers in neurorobotics* 7 (2013), p. 21.

[126] Chris J Needham et al. "A primer on learning in Bayesian networks for computational biology". In: *PLoS computational biology* 3.8 (2007), e129.

[127] Jakob Nielsen and Rolf Molich. "Heuristic evaluation of user interfaces". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1990, pp. 249–256.

[128] Michael A Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, USA, 2015.

[129] Volker Paelke et al. "User interfaces for cyber-physical systems". In: *at-Automatisierungstechnik* 63.10 (2015), pp. 833–843.

[130] Guansong Pang, Chunhua Shen, and Anton van den Hengel. "Deep anomaly detection with deviation networks". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 353–362.

[131] Guansong Pang et al. "Deep learning for anomaly detection: A review". In: *ACM computing surveys (CSUR)* 54.2 (2021), pp. 1–38.

[132] Nicolas Papernot et al. "Towards the science of security and privacy in machine learning". In: *arXiv preprint arXiv:1611.03814* (2016).

[133] Seong Hyeon Park et al. "Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 1672–1678.

[134] Animesh Patcha and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends". In: *Computer networks* 51.12 (2007), pp. 3448–3470.

[135] Judea Pearl. "Bayesian networks". In: (2011).

[136] Dragoljub Pokrajac, Aleksandar Lazarevic, and Longin Jan Latecki. "Incremental local outlier detection for data streams". In: *2007 IEEE symposium on computational intelligence and data mining*. IEEE. 2007, pp. 504–515.

[137] Samira Pouyanfar et al. "A survey on deep learning: Algorithms, techniques, and applications". In: *ACM Computing Surveys (CSUR)* 51.5 (2018), pp. 1–36.

[138] J. Ross Quinlan. "Induction of decision trees". In: *Machine learning* 1 (1986), pp. 81–106.

[139] John Ross Quinlan et al. “Inductive knowledge acquisition: a case study”. In: *Proceedings of the Second Australian Conference on Applications of expert systems*. 1987, pp. 137–156.

[140] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. “Efficient algorithms for mining outliers from large data sets”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 427–438.

[141] Francesco Ranzato and Marco Zanella. “Abstract interpretation of decision tree ensemble classifiers”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 5478–5486.

[142] Soumi Ray et al. “Using statistical anomaly detection models to find clinical decision support malfunctions”. In: *Journal of the American Medical Informatics Association* 25.7 (2018), pp. 862–871.

[143] Shebuti Rayana. “ODDS library (2016)”. In: *URL http://odds. cs. stonybrook. edu* (2016).

[144] Christian Reimers, Jakob Runge, and Joachim Denzler. “Determining the relevance of features for deep neural networks”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 330–346.

[145] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Anchors: High-precision model-agnostic explanations”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[146] N Lawrence Ricker. “Decentralized control of the Tennessee Eastman challenge process”. In: *Journal of process control* 6.4 (1996), pp. 205–221.

[147] Patrick Royston. “Multiple imputation of missing values”. In: *The Stata Journal* 4.3 (2004), pp. 227–241.

[148] Lukas Ruff et al. “Deep one-class classification”. In: *International conference on machine learning*. PMLR. 2018, pp. 4393–4402.

[149] Lukas Ruff et al. “Deep semi-supervised anomaly detection”. In: *arXiv preprint arXiv:1906.02694* (2019).

[150] Mohammad Sabokrou, Mahmood Fathy, and Mojtaba Hoseini. “Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder”. In: *Electronics Letters* 52.13 (2016), pp. 1122–1124.

[151] Franco Scarselli et al. “The graph neural network model”. In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[152] Peter Schneider and Konstantin Böttinger. “High-performance unsupervised anomaly detection for cyber-physical system networks”. In: *Proceedings of the 2018 workshop on cyber-physical systems security and privacy*. 2018, pp. 1–12.

[153] Bernhard Schölkopf et al. “Estimating the support of a high-dimensional distribution”. In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[154] Mike Schuster and Kuldip K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[155] Andrew Jhon Scott and M Knott. "A cluster analysis method for grouping means in the analysis of variance". In: *Biometrics* (1974), pp. 507–512.

[156] MA Al-Shabi. "Credit card fraud detection using autoencoder model in unbalanced datasets". In: *Journal of Advances in Mathematics and Computer Science* 33.5 (2019), pp. 1–16.

[157] Murali Shanker, Michael Y Hu, and Ming S Hung. "Effect of data standardization on neural network training". In: *Omega* 24.4 (1996), pp. 385–397.

[158] Alex Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[159] Xin Shi et al. "Early anomaly detection and localisation in distribution network: A data-driven approach". In: *IET Generation, Transmission & Distribution* 14.18 (2020), pp. 3814–3825.

[160] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[161] Kamilya Smagulova and Alex Pappachen James. "A survey on LSTM memristive neural network architectures and applications". In: *The European Physical Journal Special Topics* 228.10 (2019), pp. 2313–2324.

[162] Ralf C Staudemeyer and Eric Rothstein Morris. "Understanding LSTM–a tutorial into long short-term memory recurrent neural networks". In: *arXiv preprint arXiv:1909.09586* (2019).

[163] Per Erik Strandberg. "Automated System-Level Software Testing of Industrial Networked Embedded Systems". In: *arXiv preprint arXiv:2111.08312* (2021).

[164] Ya Su et al. "Robust anomaly detection for multivariate time series through stochastic recurrent neural network". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2828–2837.

[165] Gian Antonio Susto, Matteo Terzi, and Alessandro Beghi. "Anomaly detection approaches for semiconductor manufacturing". In: *Procedia Manufacturing* 11 (2017), pp. 2018–2024.

[166] Bayu Adhi Tama and Kyung-Hyune Rhee. "An in-depth experimental study of anomaly detection using gradient boosted machine". In: *Neural Computing and Applications* 31 (2019), pp. 955–965.

[167] Wensi Tang et al. "Rethinking 1d-cnn for time series classification: A stronger baseline". In: *arXiv preprint arXiv:2002.10061* (2020).

[168] Riccardo Taormina and Stefano Galelli. "Deep-learning approach to the detection and localization of cyber-physical attacks on water distribution systems". In: *Journal of Water Resources Planning and Management* 144.10 (2018), p. 04018065.

[169] Nesime Tatbul et al. "Precision and recall for time series". In: *Advances in neural information processing systems* 31 (2018).

[170] Kutub Thakur et al. "Impact of cyber-attacks on critical infrastructure". In: *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE. 2016, pp. 183–186.

[171] Masahito Togami et al. "Unsupervised training for deep speech source separation with Kullback-Leibler divergence based probabilistic loss function". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 56–60.

[172] Jack V Tu. "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes". In: *Journal of clinical epidemiology* 49.11 (1996), pp. 1225–1231.

[173] Imtiaz Ullah and Qusay H Mahmoud. "Design and development of RNN anomaly detection model for IoT networks". In: *IEEE Access* 10 (2022), pp. 62722–62750.

[174] Laura Uusitalo. "Advantages and challenges of Bayesian networks in environmental modelling". In: *Ecological modelling* 203.3-4 (2007), pp. 312–318.

[175] Tim Van Erven and Peter Harremos. "Rényi divergence and Kullback-Leibler divergence". In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 3797–3820.

[176] Michel Verleysen and Damien François. "The curse of dimensionality in data mining and time series prediction". In: *International work-conference on artificial neural networks*. Springer. 2005, pp. 758–770.

[177] Michael R Waldmann and Laura Martignon. "A Bayesian network model of causal learning". In: *Proceedings of the twentieth annual conference of the Cognitive Science Society*. Routledge. 1998, pp. 1102–1107.

[178] Jiafu Wan et al. "Advances in cyber-physical systems research". In: *KSII Transactions on Internet and Information Systems (TIIS)* 5.11 (2011), pp. 1891–1908.

[179] Dennis Wei, Tian Gao, and Yue Yu. "DAGs with No Fears: A closer look at continuous optimization for learning Bayesian networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3895–3906.

[180] Thomas C Williams et al. "Directed acyclic graphs: a tool for causal studies in paediatrics". In: *Pediatric research* 84.4 (2018), pp. 487–493.

[181] Kevin S Woods et al. "Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography". In: *International Journal of Pattern Recognition and Artificial Intelligence* 7.06 (1993), pp. 1417–1436.

[182] Gilberto M. Xavier and José Manoel de Seixas. "Fault Detection and Diagnosis in a Chemical Process using Long Short-Term Memory Recurrent Neural Network". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. 2018, pp. 1–8. DOI: 10.1109/IJCNN.2018.8489385.

[183] Hong-Jie Xing and Man Ji. "Robust one-class support vector machine with rescaled hinge loss function". In: *Pattern Recognition* 84 (2018), pp. 152–164.

[184] Feng Yan, Chunjie Yang, and Xinmin Zhang. "Stacked Spatial-Temporal Autoencoder for Quality Prediction in Industrial Processes". In: *IEEE Transactions on Industrial Informatics* (2022).

[185] Wenjie Yang et al. "Towards rich feature discovery with class activation maps augmentation for person re-identification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 1389–1398.

[186] Xin Yang and Dajun Feng. "Generative adversarial network based anomaly detection on the benchmark Tennessee Eastman process". In: *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE. 2019, pp. 644–648.

[187] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. "Understanding the effect of accuracy on trust in machine learning models". In: *Proceedings of the 2019 chi conference on human factors in computing systems*. 2019, pp. 1–12.

[188] Wennian Yu, Il Yong Kim, and Chris Mechefske. "Analysis of different RNN autoencoder variants for time series classification and machine prognostics". In: *Mechanical Systems and Signal Processing* 149 (2021), p. 107322.

[189] Yong Yu et al. "A review of recurrent neural networks: LSTM cells and network architectures". In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[190] Jerrold H Zar. "Spearman rank correlation". In: *Encyclopedia of biostatistics* 7 (2005).

[191] Houssam Zenati et al. "Adversarially learned anomaly detection". In: *2018 IEEE International conference on data mining (ICDM)*. IEEE. 2018, pp. 727–736.

[192] Jiuqi Elise Zhang, Di Wu, and Benoit Boulet. "Time series anomaly detection for smart grids: A survey". In: *2021 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE. 2021, pp. 125–130.

[193] Bendong Zhao et al. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169.

[194] Ming Zhao and Jingchao Chen. "A review of methods for detecting point anomalies on numerical dataset". In: *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Vol. 1. IEEE. 2020, pp. 559–565.

[195] Yue Zhao, Ryan Rossi, and Leman Akoglu. "Automatic unsupervised outlier model selection". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4489–4502.

[196] Chong Zhou and Randy C Paffenroth. "Anomaly detection with robust deep autoencoders". In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 665–674.

[197] Zongzhao Zhou et al. "Combining global and local surrogate models to accelerate evolutionary optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.1 (2006), pp. 66–76.

[198] Zahra Zohrevand and Uwe Glässer. “Should i raise the red flag? A comprehensive survey of anomaly scoring methods toward mitigating false alarms”. In: *arXiv preprint arXiv:1904.06646* (2019).