UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Anomaly Detection in Time Series Sensor Data using Statistical and Machine Learning Techniques

Diploma Thesis

Sofianos Lampropoulos

Supervisor: Dimitrios Rafailidis

April 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ανίχνευση Ανωμαλιών σε Δεδομένα Αισθητήρων Χρονοσειρών με Χρήση Στατιστικών και Αλγορίθμων Μηχανικής Μάθησης

Διπλωματική Εργασία

Σοφιανός Λαμπρόπουλος

Επιβλέπων/πουσα: Ραφαηλίδης Δημήτριος

Απρίλιος 2023

Approved by the Examination Committee:

Supervisor **Dimitrios Rafailidis**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Dimitrios Katsaros**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Michael Vasilakopoulos**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Associate Professor Dimitrios Rafailidis, for his unwavering support, guidance, and mentorship throughout my Thesis. His expertise in data sciences and applications has been invaluable, and his passion for research has been a constant source of inspiration for me. I am truly grateful for the countless hours he has spent providing insights and constructive feedback on my work.

I would also like to extend my appreciation to Associate Professor Dimitrios Katsaros for his participation in the committee for my thesis. His valuable input and expertise in distributed systems have contributed to the overall quality and integrity of my research.

Additionally, I would like to thank Professor Michael Vasilakopoulos for his insights in the areas of Data Mining, Databases, and Big Data. His knowledge and experience in these fields have significantly broadened my understanding and enriched my research work.

I am also grateful to the entire faculty and staff at the University of Thessaly for providing a stimulating and supportive academic environment in which to pursue my research. In particular, I would like to acknowledge my fellow students and colleagues, whose camaraderie and encouragement have made this journey a truly rewarding experience.

Lastly, I wish to extend my deepest thanks to my family and friends for their unwavering love, support, and understanding throughout this challenging but fulfilling endeavor. This accomplishment would not have been possible without their constant encouragement and belief in me.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Sofianos Lampropoulos

Diploma Thesis

Anomaly Detection in Time Series Sensor Data using Statistical and Machine Learning Techniques

Sofianos Lampropoulos

Abstract

The increasing availability of sensor data in various applications has led to a growing need for reliable anomaly detection techniques. This diploma thesis focuses on the development of effective anomaly detection methods for time series sensor data using both statistical and machine learning techniques. The proposed methodology includes preprocessing, feature engineering, univariate and multivariate statistical techniques, and machine learning algorithms. Specifically, we explored the use of Interquartile Range (IQR) and K-means clustering, Isolation Forest One-Class Support Vector Machines (SVM), Prophet and Long Short Term Memory (LSTM) models. We also applied Principal Component Analysis (PCA) for dimensionality reduction and SelectKBest for feature selection. The performance of these techniques was evaluated using various evaluation metrics, and their effectiveness was compared against ground truth anomalies. The results of our experiments demonstrate that machine learning algorithms, particularly LSTM, outperformed statistical methods in detecting anomalies in sensor data. The proposed techniques can be applied in various domains, such as industrial systems, internet of things (IoT), and smart cities, to detect anomalous events and improve system reliability and performance.

Keywords:

Anomaly Detection, Time Series, Sensor Data, Statistical Techniques, Machine Learning Techniques.

Διπλωματική Εργασία

Ανίχνευση Ανωμαλιών σε Δεδομένα Αισθητήρων Χρονοσειρών με Χρήση Στατιστικών και Αλγορίθμων Μηχανικής Μάθησης

Σοφιανός Λαμπρόπουλος

Περίληψη

Η αυξανόμενη διαθεσιμότητα δεδομένων αισθητήρων σε διάφορες εφαρμογές έχει οδηγήσει σε μια αυξανόμενη ανάγκη για αξιόπιστες τεχνικές ανίχνευσης ανωμαλιών. Η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάπτυξη αποτελεσματικών μεθόδων ανίχνευσης ανωμαλιών για χρονοσειρές δεδομένων αισθητήρων χρησιμοποιώντας στατιστικές τεχνικές και μεθόδους μηχανικής μάθησης. Η προτεινόμενη μεθοδολογία περιλαμβάνει προεπεξεργασία, μηχανική χαρακτηριστικών, μονομεταβλητές και πολυμεταβλητές στατιστικές τεχνικές και αλγόριθμους μηχανικής μάθησης. Συγκεκριμένα, διερευνήσαμε τη χρήση των Interquartile Range (IQR), K-means συσταδοποίηση, Isolation Forest, Μηχανές διανυσμάτων υποστήριξης μίας κλάσης (SVM), Prophet και Μοντέλα Μακροχρόνιας Βραχυπρόθεσμης Μνήμης (LSTM). Εφαρμόσαμε επίσης ανάλυση κύριων συνιστωσών (PCA) για τη μείωση της διαστατικότητας και SelectKBest για την επιλογή χαρακτηριστικών. Τα αποτελέσματα των πειραμάτων μας καταδεικνύουν ότι οι αλγόριθμοι μηχανικής μάθησης, ιδίως ο LSTM, υπερτερούν των στατιστικών μεθόδων στον εντοπισμό ανωμαλιών σε δεδομένα αισθητήρων. Οι προτεινόμενες τεχνικές μπορούν να εφαρμοστούν σε διάφορους τομείς, όπως τα βιομηχανικά συστήματα, το διαδίκτυο των πραγμάτων (IoT) και τις έξυπνες πόλεις, για την ανίχνευση ανώμαλων συμβάντων και τη βελτίωση της αξιοπιστίας και της απόδοσης των συστημάτων.

Λέξεις-κλειδιά:

Ανίχνευση ανωμαλιών, χρονοσειρές, δεδομένα αισθητήρων, στατιστικές τεχνικές, τεχνικές μηχανικής μάθησης.

Table of contents

List of figures

List of tables

Abbreviations

AEI Ανώτατο Εκπαιδευτικό Ίδρυμα

BPF Band Pass Filter

LSTM Long Short-Term Memory

RNN Recurrent Neural Network

ADF Augmented Dickey-Fuller

IoT Internet of Things

PCA Principal Component Analysis

OCSVM One Class Support Vector Machine

KNN K-Nearest Neighbors

VAE Variational Autoencoder

HMM Hidden Markov Model

ROC Receiver Operating Characteristic

AUC Area Under the Curve

RMSE Root Mean Squared Error

SAX Symbolic Aggregate approximation

ADS Anomaly Detection System

IF Isolation Forest

LOF Local Outlier Factor

Chapter 1

Introduction

The modern world is witnessing rapid advancements in technology and data collection, transforming the way we perceive and interact with our surroundings. In this context, analyzing and understanding complex systems has become increasingly important for addressing challenges and ensuring optimal performance. The site under investigation in this thesis embodies a combination of intricate processes, where multiple sensors and subsystems are interconnected, generating a wealth of time series data. This data holds valuable insights into the underlying dynamics and potential anomalies of the site.

The site in question features a diverse range of components, each contributing to the overall functionality and efficiency of the system. These components interact with one another, forming a complex network where the behavior of individual elements can significantly impact the system as a whole. Understanding the relationships between these components, as well as the potential sources of anomalies, is crucial for the continued success of the site's operations.

As the site operates, it faces a multitude of challenges that can compromise its efficiency, reliability, and safety. These challenges arise from various sources, such as equipment malfunctions, process disturbances, and unexpected external factors. The manifestation of these issues can lead to suboptimal performance or even catastrophic failures, resulting in significant financial and operational losses.

In light of these challenges, this thesis aims to investigate the site's complexities and develop a comprehensive understanding of the factors that contribute to its performance. By delving into the wealth of sensor data generated by the site, we will identify patterns, trends, and anomalies that can provide valuable insights into its inner workings. Through

this analysis, we will uncover the root causes of the site's challenges and propose solutions that can enhance its performance and resilience.

The subsequent chapters of this thesis will explore the various dimensions of the site, including its components, subsystems, and the relationships between them. We will delve into the methodologies and techniques employed to analyze the sensor data, culminating in the development of a novel approach to anomaly detection. Ultimately, this thesis aims to provide a comprehensive understanding of the site's complexities and offer practical solutions for addressing the challenges it faces.

1.1 Time Series Sensor Anomaly Detection Use Cases in Preventive Maintenance

Anomaly detection in time series sensor data plays a crucial role in various applications, particularly in the context of preventive maintenance. By identifying deviations from expected behavior, it is possible to take proactive measures to address potential issues before they lead to costly downtime or equipment failure. In this section, we discuss several use cases where time series sensor anomaly detection is employed for preventive maintenance across various industries and applications.

- **Industrial Equipment Monitoring**: In manufacturing and production environments, continuous monitoring of equipment, such as motors, pumps, and conveyor systems, is critical for ensuring smooth operations [1]. Anomaly detection in sensor data can help identify early signs of wear, mechanical stress, or other issues, enabling timely maintenance interventions to prevent equipment failure and production downtime [2].

- **Energy and Utilities**: In the energy sector, time series sensor data is collected from power generation equipment, such as turbines, generators, and transformers, to monitor their performance and health [3]. Anomaly detection techniques can identify unusual patterns or trends, signaling potential issues that may require maintenance or inspection, ultimately improving the reliability and efficiency of energy production and distribution systems [4].

- **Transportation**: Vehicles and transportation systems, such as trains, aircraft, and ships, rely on an array of sensors to monitor various components and subsystems [5]. Time

series anomaly detection can help detect emerging issues, such as engine performance degradation, structural fatigue, or control system malfunctions, allowing for targeted maintenance actions to ensure safe and efficient operations [6].

- **Smart Cities and Infrastructure**: Modern cities are increasingly relying on sensor networks to monitor the performance and condition of critical infrastructure, such as bridges, tunnels, and water supply systems [7]. Anomaly detection techniques applied to time series sensor data can identify potential issues, enabling proactive maintenance and reducing the risk of infrastructure failure [8].

- **Environmental Monitoring**: Sensor networks are also employed for monitoring air quality, water levels, and other environmental factors [9]. Anomaly detection in time series data can help identify unusual patterns or events, such as pollution spikes, flooding risks, or equipment malfunctions, which may require intervention or maintenance actions to mitigate potential hazards [10].

- **Healthcare and Medical Devices**: Time series data from wearable sensors and medical devices can be analyzed for anomaly detection to identify irregular patterns or trends in physiological signals, such as heart rate, blood pressure, or glucose levels [11]. These insights can help healthcare providers make informed decisions about patient care and preventive measures, as well as aid in the maintenance of medical equipment [12].

- **Internet of Things (IoT) Devices**: The widespread adoption of IoT devices has led to the generation of vast amounts of sensor data across various applications, such as home automation, security, and agriculture [13]. Anomaly detection techniques can be applied to identify unusual behaviors, security threats, or equipment malfunctions, facilitating proactive maintenance and improving overall system performance [14].

  These use cases illustrate the wide-ranging applications of time series sensor anomaly detection for preventive maintenance across diverse sectors. The development of effective anomaly detection techniques can significantly contribute to the enhancement of system reliability, efficiency, and safety in these domains.

1.2 Contribution

In this thesis, we make several key contributions to the field of time series sensor anomaly detection, which are summarized below:

1. Development of an integrated framework for anomaly detection that integrates preprocessing, feature engineering, dimensionality reduction, and multiple detection methods.

2. Propose a new model with LSTM (ADS-LSTM) that exploits the advantages of LSTM networks for handling time series data and addresses their limitations in anomaly detection.

3. Extensive evaluation of the proposed ADS-LSTM model and comparison with other state-of-the-art methods using a real sensor dataset from a manufacturing company, demonstrating the superior performance of our approach in terms of accuracy, recall, F1-score and AUC.

4. In-depth analysis of the factors affecting the performance of different anomaly detection techniques, providing insights and recommendations for practitioners working with time series sensor data.

5. Identify potential areas for future work and improvements, paving the way for further progress in the field of time series sensor anomaly detection and predictive maintenance.

1.3 Thesis Outline

This thesis is organized into five chapters, providing a comprehensive study of time series sensor anomaly detection and its applications in preventive maintenance. The chapters are structured as follows:

- **Chapter 1: Introduction** - This chapter introduces the topic of time series sensor anomaly detection, its importance, and its use cases in preventive maintenance. It also presents the main contributions of the thesis.
- **Chapter 2: Literature Review** - This chapter reviews existing literature and techniques in the field of time series sensor anomaly detection, covering statistical methods, machine learning techniques, and deep learning techniques. It also discusses the two main types of anomaly detection: point-based and subsequence-based approaches.
- **Chapter 3: Proposed Model** - In this chapter, we present our proposed Anomaly Detection System using LSTM (ADS-LSTM) model. We describe the model's input-output, LSTMs architecture, and training process in detail.
- **Chapter 4: Experiments** - This chapter describes the experiments conducted to evaluate the performance of our proposed ADS-LSTM model and compares it with other existing methods. We present the sensor dataset used, the evaluation metrics, the compared methods, and the experimental results. Visualization of the results is also provided to better understand the performance of each method.
- **Chapter 5: Conclusion** - This chapter summarizes the main findings and contributions of the thesis. We discuss the strengths and limitations of our proposed ADS-LSTM model and reflect on its performance compared to other methods. We also provide insights into the practical implications of our research and suggest potential areas for future work in the field of time series sensor anomaly detection. Finally, we conclude the thesis by emphasizing the importance of accurate anomaly detection in preventive maintenance and its potential impact on industrial systems.

Chapter 2

Literature Review

The field of time series sensor anomaly detection encompasses a diverse range of methods and techniques, originating from traditional statistical approaches and evolving towards contemporary machine learning and deep learning algorithms. In this chapter, we present a comprehensive review of the key methodologies and techniques employed for detecting anomalies in time series sensor data. Furthermore, we discuss the classification of these techniques into two primary categories: ***point-based*** anomaly detection (Section 2.2) and ***subsequence-based*** anomaly detection (Section 2.3).

The ***point-based*** anomaly detection techniques focus on the identification of individual data points that exhibit significant deviations from the expected patterns or values in the time series. These techniques are particularly applicable to sensor data, where an isolated anomalous reading may signify a malfunctioning sensor or an event warranting immediate attention.

Conversely, the ***subsequence-based*** anomaly detection techniques are concerned with detecting anomalous patterns within subsequences of the time series. The primary objective of these techniques is to uncover segments of the time series characterized by unusual behavior, rather than pinpointing individual data points. Subsequence-based anomaly detection is especially relevant for applications in which temporal relationships between data points are of paramount importance, such as identifying atypical patterns of activity in financial time series or detecting anomalous sequences in biological data. In this chapter, we provide an overview of the key methods and techniques that have been utilized for detecting anomalies in time series sensor data, and discuss their categorization into two main types: ***point-based*** anomaly detection (Section 2.2) and ***subsequence-based*** anomaly detection (Section 2.3).

2.1 Methods and Techniques for Time Series Sensor Anomaly Detection

The domain of time series sensor anomaly detection has seen the development and application of various methods and techniques, spanning from traditional statistical approaches to modern machine learning and deep learning algorithms. In this section, we provide an overview of the key methods and techniques that have been utilized for detecting anomalies in time series sensor data.

2.1.1 Statistical Methods

1. **Moving Average**: This technique involves calculating the average value of data points within a sliding window to smooth out short-term fluctuations and highlight long-term trends [15]. Anomalies are identified when the observed values deviate significantly from the moving average.

2. **Exponential Smoothing**: Similar to moving averages, exponential smoothing applies a weighted average to the time series data, with more recent data points receiving higher weights [16]. This method can adapt more quickly to changes in the underlying patterns and detect anomalies more effectively.

3. Autoregressive Integrated Moving Average (**ARIMA**): ARIMA is a widely used statistical method for modeling and forecasting time series data [17]. It can be employed for anomaly detection by comparing observed values to the predicted values generated by the ARIMA model and identifying significant deviations as potential anomalies.

4. Seasonal Decomposition of Time Series (**STL**): STL is a technique for decomposing time series data into its trend, seasonal, and residual components [18]. By analyzing the residual component, it is possible to detect anomalies that do not conform to the regular pattern of the time series.

5. Hidden Markov Models (**HMM**): HMMs are statistical models used to describe sequences of observations [19]. They can be applied to anomaly detection by modeling the normal behavior of a system and identifying data points that have low likelihood under the learned model.

2.1.2 Machine Learning Techniques

1. **Clustering** Algorithms: Unsupervised learning techniques, such as K-means clustering [20] and DBSCAN [21], can be applied to group similar data points together. Anomalies are then identified as data points that do not belong to any cluster or are significantly distant from their nearest cluster centroids.

2. **PCA**: PCA is a dimensionality reduction technique that can be employed for anomaly detection by projecting the high-dimensional sensor data onto a lower-dimensional space [22]. Outliers in the reduced space can be considered potential anomalies.

3. **Isolation Forest**: This ensemble-based method works by isolating data points using random partitioning of the feature space [23]. Anomalies are typically isolated more quickly than normal data points, and their isolation scores can be used to identify potential outliers.

4. **OCSVM**: One-Class SVM is an unsupervised learning algorithm that constructs a decision boundary around the majority of the data points in the feature space [24]. Data points outside this boundary are considered anomalies.

5. **LOF**: LOF is an unsupervised method that computes the local density deviation of a given data point compared to its neighbors [25]. Data points with significantly lower density than their neighbors are considered outliers.

6. **k-NN**: k-NN is a non-parametric method used for classification and regression [26]. In the context of anomaly detection, k-NN can be used to measure the distance between a data point and its k-nearest neighbors. Data points with large distances to their neighbors are considered anomalies.

7. **Prophet**: Developed by Facebook, the Prophet model is a time-series forecasting technique that can also be used for anomaly detection. It works by decomposing time series data into its seasonal, trend, and residual components. By comparing the observed values with the model's predictions, anomalies can be identified as data points that deviate significantly from the expected values.

2.1.3 Deep Learning Techniques

1. **Autoencoders**: Autoencoders are unsupervised neural networks that learn to encode and reconstruct input data [27]. For anomaly detection, the reconstruction error between the input data and the reconstructed output can be used as an indicator of how well the data conforms to the learned patterns, with high error rates suggesting potential anomalies [28].

2. **LSTM** Networks: LSTMs are a type of recurrent neural network (RNN) specifically designed for handling time series data [29]. They can be used for anomaly detection by training the network to predict future values based on historical data, and then comparing the predicted values to the actual values to detect significant deviations [30].

3. **VAE**: VAEs are generative models that learn a probabilistic mapping between the data and a latent space [31]. By analyzing the likelihood of data points in the latent space, VAEs can be used to identify anomalies that do not conform to the learned distribution [32].

2.2 Point-based Anomaly Detection

In recent years, numerous studies have addressed the problem of anomaly detection in point-based sensor data time series. In this section, we present a selection of 12 notable works that have contributed to progress in this area.

Lazarevic et al. (2005)[33] present a technique called ”feature bagging”, which combines the results of multiple outlier detection algorithms to improve the overall performance in detecting anomalies in sensor data points.

Chandola et al. (2009) [34] provided a comprehensive overview of anomaly detection techniques in their seminal work, including applications to sensor data time series. This study laid the foundation for many subsequent studies in the field. Hochenbaum et al. (2017) [35] proposed a statistical learning-based approach for automatic anomaly detection in cloud-based time series data, demonstrating its applicability to large-scale sensor data.

Malhotra et al. (2015) [36] presented an encoder-decoder architecture based on long short-term memory (LSTM) for anomaly detection in multivariate sensor data, demonstrating the potential of deep learning methods in this area. In their work, Zhang et al. (2018) [37] pro-

posed a deep learning approach for unsupervised ensemble learning for anomaly detection in time series sensor data, highlighting the benefits of combining multiple models.

Ahmed et al. (2016) [38] conducted a review of several anomaly detection techniques in networks, including those based on time series sensor data. Their work provided valuable insights into the challenges and opportunities in this area. Laptev et al. (2015) [39] introduced a general and scalable framework for automated time series anomaly detection, which can be applied to sensor data in a wide range of domains.

In addition to the previously mentioned studies, many other works have made significant contributions to the field of anomaly detection in sensor data time series at points Susto et al. (2014) [40] applied unsupervised machine learning techniques, such as clustering and principal component analysis, to anomaly detection for sensor data. Goldstein and Uchida (2016)[41] conducted a comparative evaluation of several unsupervised anomaly detection algorithms, including principal component analysis (PCA), one-class SVM, and local outlier factor (LOF), applied to multivariate sensor data.

Feng et al. (2017)[42] addressed the problem of robust anomaly detection in the presence of sensor noise using a combination of statistical and machine learning techniques. In their study, Sakurada and Yairi (2014) [28] proposed a new anomaly detection technique that uses auto-encoders, a type of artificial neural network, to reduce the dimensionality of sensor data. Kieu et al. (2018) [43] introduced an unsupervised deep learning method to detect anomalies in time series sensor data collected from cyber-physical systems (CPS).

Collectively, these studies demonstrate the wide range of techniques and algorithms that have been applied to anomaly detection in time series sensor data, from unsupervised machine learning to deep learning approaches. Continued development and improvement of these methods will undoubtedly lead to more accurate and reliable anomaly detection systems for various applications.

2.3 Subsequence-based Anomaly Detection

In this section, we discuss notable work that focuses specifically on detecting time series anomalies in time subsequences. Keogh et al. (2005)[44] introduced the HOT SAX algorithm for efficient detection of unusual subsequences in time series data, demonstrating its effectiveness for anomaly detection in various applications. Lin et al. (2007)[45] proposed the Symbolic Aggregate approximation (SAX) method for time series data representation, which enables efficient processing and analysis of time subsequences.

Buono et al. (2007) presented a visual analysis approach for anomaly detection in time subsequences, highlighting the importance of human interpretation in the anomaly detection process. Wei et al. (2005)[46] investigated nearest-neighbor-based feature selection for regression, demonstrating its applicability to anomaly detection in temporal subsequences of neural activity data. Zhu and Shasha (2003) [47] introduced a method for indexing time series subsequences using envelope transformations, allowing efficient retrieval and comparison of subsequences for anomaly detection.

Goldin and Uretsky (2007)[48] proposed a learning automatic-based approach for discovering temporal patterns in multivariate time series, which can be used for anomaly detection in time subsequences. Gupta et al. (2010)[49] introduced a method for detecting heterogeneous changes in time series subsequences, which can be applied to various domains for anomaly detection. Finally, Rakthanmanon et al. (2012) [50] presented an efficient approach for searching and mining trillions of time series subsequences under dynamic time warping (DTW), demonstrating its usefulness for anomaly detection and other applications.

Yeh et al. (2016) [51] proposed a matrix profile-based approach for discovering time series motifs and discords, which can be used for detecting anomalies in time subsequences. The matrix profile enables fast and accurate identification of unusual patterns in time series data, outperforming existing methods in terms of both computational efficiency and accuracy.

Senin et al. (2018) [52] introduced the GrammarViz algorithm, which utilizes a grammar-based approach to represent and mine time series subsequences. This method allows for the efficient discovery of unusual patterns in time series data, enabling accurate anomaly detection in various applications.

Wang et al. (2013) [53] presented a method for detecting anomalous subsequences in time series using variable-length Markov chains. This approach leverages the statistical properties of subsequences and can be applied to a wide range of domains, including finance, health,

and environmental monitoring.

Vahdatpour et al. (2009) [54] investigated the use of motif-based analysis for detecting anomalies in human activity data collected from wearable sensors. Their work demonstrated the effectiveness of subsequence-based anomaly detection for identifying unusual patterns in complex, real-world time series data.

Chakrabarti et al. (2002) [55] introduced the concept of ”locality-sensitive hashing” for efficient subsequence matching in time series databases. This technique enables fast retrieval and comparison of subsequences, significantly reducing the computational complexity of anomaly detection tasks.

These studies highlight the variety of techniques and approaches developed for anomaly detection in time subsequences, ranging from efficient indexing methods to visual analysis and strategies based on automation learning.

Chapter 3

Proposed Model

In this section, we propose our model, **ADS-LSTM**, to address the problem of anomaly detection using data from three selected sensors. The model is based on three different LSTM architectures, each of which is tailored to one of the three sensors identified through the Univariate Feature Selection (SelectKbest).

We selected these sensors based on the Univariate Feature Selection (SelectKbest) method, which evaluates the individual importance of each sensor by comparing their contribution to the target variable (machine status). This method allows us to focus on the most informative sensors for anomaly detection, reducing the complexity of the model and improving its computational efficiency. The three selected sensors showed the highest scores in the feature selection process, indicating their strong importance in anomaly detection.

We also selected the LSTM architecture for our model due to its ability to efficiently capture long-term dependencies in time series data. It has been shown that LSTMs overcome the vanishing gradient problem commonly faced by traditional recurrent neural networks (RNNs), allowing them to learn complex patterns over extended periods. This feature makes LSTMs suitable for anomaly detection tasks in sensor data, where temporal relationships can be critical for detecting abnormal events. Several studies in the literature, such as Malhotra et al. (2015), have demonstrated the effectiveness of LSTMs for anomaly detection in multivariate sensor data.

3.1 The ADS-LSTM Model

The ADS-LSTM model consists of the following components:

3.1.1 Input - Output

The input to the ADS-LSTM model consists of time series data from three selected sensors. Each LSTM architecture is specifically designed to handle data from one of these sensors. The input scheme for each LSTM is $(TIME\_STEPS, 1) \equiv (10, 1)$, where TIME_STEPS represents the number of consecutive data points or time intervals considered in the input sequence.

The output generated by each LSTM model corresponds to a binary classification, indicating whether the associated sensor data contains an anomaly or not. To consolidate the outputs from the three LSTM models and provide a comprehensive evaluation of the entire system, a Majority Voting Ensemble mechanism is employed. This approach takes into account the information gathered from all three selected sensors and yields a final binary classification for the entire system, signifying the presence or absence of an anomaly. In this way, the input and output mechanisms of the ADS-LSTM model work together to effectively detect anomalies in the time series sensor data.

3.1.2 The LSTM Architecture

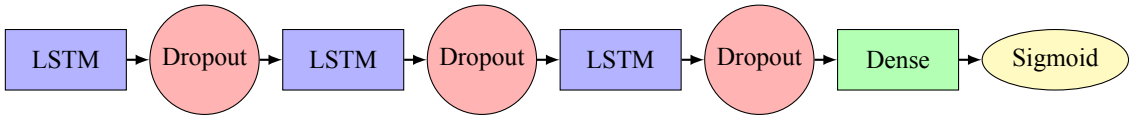

Figure 3.1: Flowchart of the LSTM architecture with multiple LSTM, Dropout, Dense, and Activation layers.

The LSTM (Long Short-Term Memory) architecture used in this study consists of several layers arranged sequentially, as shown in Figure 3.1. These layers are designed to process time series data, learn intricate patterns, and address the vanishing gradient problem commonly encountered in traditional Recurrent Neural Networks (RNNs). In this section, we describe

the function of each layer and the associated mathematical operations.

1. **LSTM Layers:** The architecture features three LSTM layers with 64, 256, and 100 hidden units, respectively. These layers are designed to learn short- and long-term dependencies in the input sequence and capture higher-level patterns. Mathematically, the LSTM layers consist of the following components:

   - Input gate: $i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$
   - Forget gate: $f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$
   - Cell state update: $\tilde{c}t = \tanh(Wigx_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$
   - Output gate: $o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$
   - Cell state: $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
   - Hidden state: $h_t = o_t \odot \tanh(c_t)$

   where x_t is the input at time step t, h_{t-1} is the hidden state from the previous time step, W and b are the weight matrices and bias vectors for each gate, σ is the sigmoid function, and $\odot$ represents element-wise multiplication.

2. **Dropout Layers:** To prevent overfitting, Dropout layers are applied after each LSTM layer. They randomly set a fraction of the input units to 0 during training. The rate of dropout is set to 0.2, meaning that 20% of the input units are dropped. Mathematically, the Dropout layers can be represented as:

   $$y_t = x_t \odot m_t$$

   where y_t is the output at time step t, x_t is the input, m_t is a binary mask vector with the same shape as x_t, and $\odot$ is element-wise multiplication.

3. **Dense Layer :** The Dense layer is a fully connected layer that takes the output from the previous LSTM layer and computes a weighted sum of its input units. It is followed by a bias term and an activation function. Mathematically, the Dense layer can be represented as:

$$y = Wx + b$$

where y is the output, W is the weight matrix, x is the input, and b is the bias vector.

4. **Activation Layer:** The Activation layer applies the sigmoid function to the output of the Dense layer, which is used to transform the result into a probability value between 0 and 1. Mathematically, the sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

where x is the input and $\sigma(x)$ is the output.

In summary, the LSTM architecture processes time series data through multiple LSTM, Dropout, Dense, and Activation layers to learn complex patterns and produce a final binary classification result, indicating the presence or absence of an anomaly.

3.1.3 Training Process

The training process of each LSTM model involves several mathematical concepts and techniques, which we will discuss in this section.

LSTMs are a type of recurrent neural network (RNN) designed to learn and retain long-term dependencies in sequential data. As we discussed before, the LSTM cell consists of an input gate i_t, a forget gate f_t, an output gate o_t, and a cell state c_t. The LSTM cell updates its state and produces an output h_t at each time step t based on the input x_t, the previous hidden state h_{t-1}, and the previous cell state c_{t-1}.

Each LSTM model is trained using the following steps:

1. Initialize the weight matrices and bias vectors for the input, forget, and output gates, as well as the cell state.

2. For each epoch, perform the following steps:

   (α′) For each mini-batch in the training data, apply the forward pass, computing the hidden states, cell states, and output of the LSTM layers.

   (β′) Calculate the binary cross-entropy loss for the mini-batch, taking into account the class weights.

$$L = -\sum_{i=1}^{N} w_i [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{3.1}$$

   (γ′) Compute the gradients of the loss function with respect to the LSTM weights and biases using the BPTT algorithm.

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial W} \tag{3.2}$$

   (δ′) Update the weights and biases using the RMSprop optimization algorithm.

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho) g_t^2 \tag{3.3}$$

$$W_t = W_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{3.4}$$

3. After training for a specified number of epochs or until the early stopping criterion is met, the model is considered trained.

4. Once all three LSTM models are trained, their results are combined using the Majority Vote operation to produce the final binary classification. In the Majority Vote operation, the class that receives the most votes from the three LSTM models is selected as the final prediction.The Majority Vote operation combines the binary classification predictions of the three LSTM models, $y^{(1)}$, $y^{(2)}$, and $y^{(3)}$. The final prediction $y^{(M)}$ is determined by selecting the class with the most votes. Mathematically, the Majority Vote operation can be represented as follows:

$$y^{(M)} = \begin{cases} 1, & \text{if } \sum_{i=1}^{3} y^{(i)} \geq 2 \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

Here, $y^{(M)}$ is the final prediction, and $y^{(i)}$ represents the binary prediction from the i-th LSTM model. If the sum of the predictions from the three LSTM models is greater than or equal to 2, the Majority Vote operation predicts class 1. Otherwise, it predicts class 0.

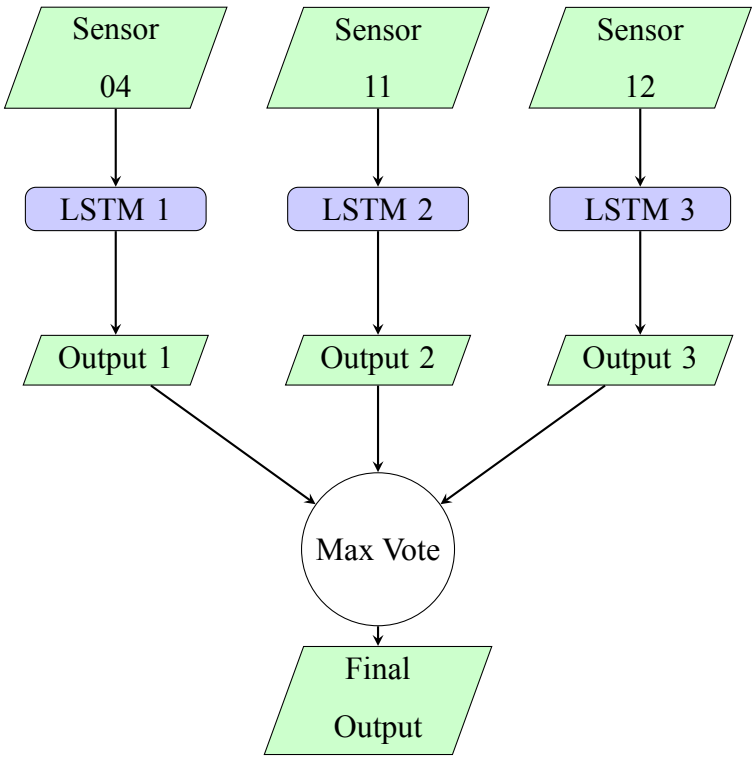

Figure 3.2: ADS-LSTM architecture with three separate LSTMs for each sensor and their results are combined using a Majority Vote operation to produce the final binary classification.

Chapter 4

Experiments

In this section, we detail the experimental setup used to evaluate the performance of our proposed ADS-LSTM model for anomaly detection. We first provide an overview of the dataset used for our experiments, followed by a description of the preprocessing steps and the evaluation metrics employed.

4.1 Sensor Dataset Statistics and Description

The dataset used in this thesis, shared by a manufacturing company on Kaggle to encourage data engineers and researchers to address the problem of anomaly detection and propose the best solution for accurately predicting the ground truth label of machine_status, contains sensor measurements from 53 sensors installed in a pump to measure various behaviors. With 220,320 observations collected in one-minute intervals over an extended duration from April 2018 to September 2018, each observation includes a timestamp, readings from the 53 sensors, and the machine_status column, which serves as a ground truth label, indicating whether the pump is operating under normal conditions or experiencing an anomaly at a given time. The rich sensor data provided by the manufacturer, comprising time series data for variables such as temperature, pressure, vibration, and flow rate, encapsulates both standard operating conditions and meticulously annotated anomalous events corresponding to diverse categories of faults or malfunctions within the industrial system. This data facilitates the development and evaluation of various anomaly detection techniques, ultimately helping to improve predictive maintenance strategies for industrial systems by accurately identifying anomalies in pump operation, allowing potential problems to be proactively addressed, minimizing the

risk of unexpected failures, and reducing maintenance costs.

Below, we provide some key statistics about the dataset:

- **Total number of data points:** The dataset comprises a total of 220,521 data points, where 220,521 is the number of observations collected during the entire monitoring period.

- **Number of sensors:** The data was collected from 51 different sensors, each measuring a specific variable related to the system's operation.

- **Time resolution:** The dataset has a time resolution of 1 minute, which means that sensor measurements were recorded at regular intervals of 1 minute.

- **Machine status distribution:** The machine status distribution in the dataset is as follows: NORMAL (205,067), RECOVERING (14,447), and BROKEN (7).

- **Anomaly distribution:** Considering both RECOVERING and BROKEN states as non-normal, there are a total of 14,454 anomalous events in the dataset. These anomalies are distributed unevenly across the dataset, with some periods having a higher concentration of faults than others.

In our analysis, we start by examining the dataset's descriptive statistics to gain an understanding of the sensor data distribution and identify potential outliers or trends. We then visually inspect the time series data for each sensor to assess its stationarity and detect any underlying trends or seasonality that may affect model performance. Next, we use Principal Component Analysis (PCA) to perform dimensionality reduction on the standardized dataset, preserving the maximum amount of variance and determining an optimal number of principal components (PCs) based on inertia. To ensure the stationarity and independence of the PCs, we conduct the Augmented Dickey-Fuller (ADF) test and compute their autocorrelation, which confirms their appropriateness for inclusion in the model. Finally, we employ univariate feature selection to identify the most relevant features that significantly contribute to the model's ability to differentiate between standard and anomalous operating conditions, as shown in the table.

Based on the feature selection, we find the following top three features:

Sensor	Chi-squared Score
sensor_04	10,106.76
sensor_11	9,879.05
sensor_12	8,167.18

Table 4.1: Top three features with the highest chi-squared scores.

In conclusion, the exploratory data analysis process includes data preprocessing, dimensionality reduction, stationarity and autocorrelation assessments, and feature selection to ensure the appropriateness of the selected features for modeling and the effective differentiation of normal and abnormal machine operation conditions.

4.2 Evaluation Metrics

In this section, we describe the evaluation metrics used to assess the performance of our anomaly detection models. These metrics allow us to objectively compare different models and help us choose the most suitable one for our application. We utilized the following evaluation metrics: Accuracy, Precision, Recall, F1-score, and Area Under the Curve (AUC).

4.2.1 Accuracy

Accuracy is a widely used performance metric in classification problems. It measures the proportion of correctly classified instances out of the total number of instances. Accuracy can be mathematically defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{4.1}$$

While accuracy is a useful metric, it can be misleading when the class distribution is imbalanced, as it might favor the majority class. In our case, anomalies represent a small fraction of the total observations, and thus, we also consider other metrics to obtain a more comprehensive evaluation.

4.2.2 Precision

Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It can be mathematically defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.2}$$

Precision is particularly useful when the cost of false positives is high. In our case, a false positive would indicate an incorrect identification of an anomaly, which could lead to unnecessary maintenance or downtime.

4.2.3 Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of true positive predictions out of all actual positive instances. It can be mathematically defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.3}$$

Recall is particularly useful when the cost of false negatives is high. In our case, a false negative would mean failing to detect an anomaly, which could result in equipment failure or significant damage.

4.2.4 F1-score

F1-score is the harmonic mean of precision and recall and provides a single measure that balances the trade-off between these two metrics. It can be mathematically defined as:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

The F1-score ranges from 0 to 1, with higher values indicating better performance. It is particularly useful when dealing with imbalanced datasets, as it takes both false positives and false negatives into account.

4.2.5 Area Under the Curve (AUC)

The Area Under the Curve (AUC) is a performance metric for binary classification problems that measures the ability of a model to distinguish between positive and negative classes. The AUC is calculated from the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (recall) against the false positive rate.

AUC values range from 0 to 1, with an AUC of 0.5 representing a random classifier and an AUC of 1 representing a perfect classifier. A higher AUC value indicates a better model in terms of its ability to differentiate between the two classes.

In summary, we used a combination of accuracy, precision, recall, F1-score, and AUC to evaluate the performance of our anomaly detection models. These metrics provide a comprehensive assessment of the model's ability to correctly identify anomalies while minimizing false positives and false negatives.

4.3 Compared Methods

In this section, we provide an overview of the anomaly detection methods we employed and compared in our study, including our proposed ADS-LSTM model. The methods are:

- **Interquartile Range (IQR)**: A statistical method used to identify outliers in the data by calculating the range between the first quartile (25th percentile) and the third quartile (75th percentile) of the data distribution. Data points that fall below or above a predefined threshold, typically 1.5 times the IQR, are considered outliers or anomalies.

- **K-means**: An unsupervised clustering algorithm that partitions the data into K clusters based on their similarity. In the context of anomaly detection, data points that have high distances from their cluster centroids can be considered anomalies.

- **Isolation Forest**: An unsupervised tree-based ensemble method for anomaly detection. It isolates data points by randomly selecting features and splitting values, with the assumption that anomalies can be isolated more quickly than normal instances. Anomalies are identified based on their average path lengths in the trees.

- **One Class SVM**: An unsupervised algorithm that identifies anomalies by learning a decision boundary around the normal data instances. Data points that fall outside this boundary are considered anomalies.

- **Prophet**: A time series forecasting model developed by Facebook that can be used for anomaly detection. By fitting the model to the historical data and forecasting future values, we can identify anomalies as deviations from the predicted values.

- **Anomaly Detection System using LSTM (ADS-LSTM)**: Our proposed model, which is an extension of the LSTM model. It incorporates additional features and techniques to improve the performance of the LSTM model in detecting anomalies. By leveraging the strengths of the LSTM architecture, ADS-LSTM aims to achieve better detection accuracy while minimizing false positives and false negatives.

In conclusion, we compare a variety of methods, ranging from statistical techniques to advanced machine learning models, in order to evaluate their effectiveness in detecting anomalies in our dataset. This comprehensive comparison allows us to identify the most suitable method for our specific application and provides insights into the strengths and weaknesses of each approach.

Model	**Functionality**
IQR	Identifies outliers based on the range between the 25th and 75th percentiles of the data distribution.
K-means	Clusters data into groups based on similarity and detects anomalies as points distant from their cluster centroids.
Isolation Forest	Isolates data points using a tree-based ensemble method and identifies anomalies based on their average path lengths in the trees.
One Class SVM	Learns a decision boundary around normal instances and considers points outside this boundary as anomalies.
Prophet	Forecasts time series data and detects anomalies as deviations from predicted values.
ADS-LSTM	An extension of the LSTM model, incorporating additional features and techniques to improve anomaly detection performance.

Table 4.2: Overview of compared anomaly detection methods and their functionalities.

4.4 Performance Evaluation Metrics

In this section, we present the performance evaluation metrics of the compared anomaly detection methods, including our proposed ADS-LSTM model. The evaluation metrics used are Accuracy, Precision, Recall, F1-score, AUC (Area Under the Receiver Operating Characteristic curve). The results are summarized in Table 1.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**AUC**
IQR	0.811312	0.873332	0.811312	0.840448	0.481033
K-means	0.820486	0.876358	0.820486	0.846697	0.496779
Isolation Forest	0.820755	0.876561	0.820755	0.846923	0.497824
One Class SVM	0.896629	0.883773	0.896629	0.890004	0.523997
Prophet	0.930007	0.930387	0.930007	0.930196	0.718228
ADS-LSTM	0.965792	0.958281	0.965792	0.96424	0.971913

Table 4.3: Performance Evaluation Metrics

As can be observed from the table, the ADS-LSTM model outperforms all other methods in terms of all evaluation metrics, achieving an accuracy of 96.57%, a precision of 95%, a recall of 96%, an F1-score of 96%, an AUC of 97%. This demonstrates the effectiveness of using LSTM models for anomaly detection tasks in time series data. It is worth noting that the Prophet model also shows competitive performance, especially when compared to the other traditional methods.

The results indicate that LSTM-based models, like the proposed ADS-LSTM, can be highly effective in detecting anomalies in time series data, outperforming a variety of other methods. This highlights the importance of leveraging advanced machine learning techniques for anomaly detection tasks, especially when dealing with complex and high-dimensional sensor data.

4.5 Experimental Visualization

In this section, we present the experimental figures obtained from the compared methods and discuss the visualization techniques employed to better understand the results. We generate a total of 18 figures (3 figures for each of the 6 models) to provide a comprehensive representation of the model performances. The figures were created using the Matplotlib and Prophet libraries for data visualization.

4.5.1 Model 1 : Interquantile Range

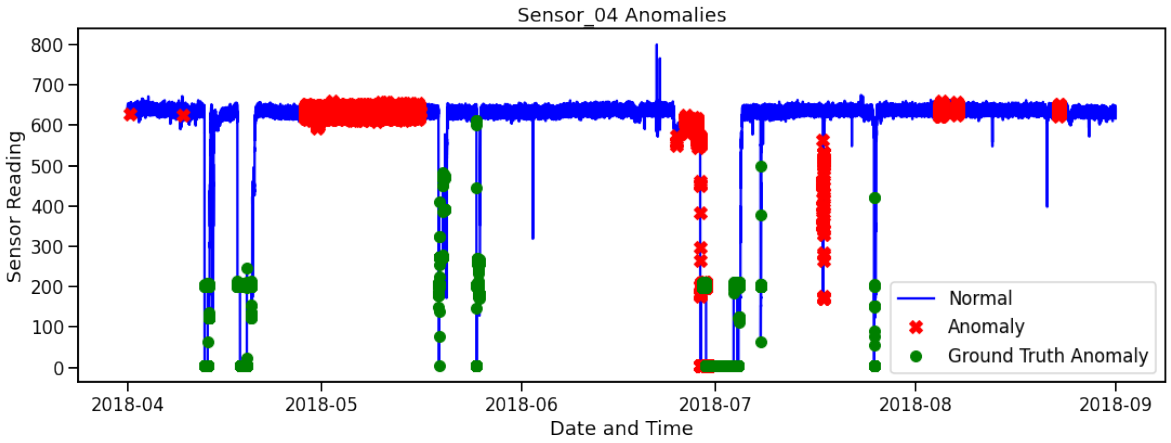

Figure 4.1: Sensor 04 Interquantile Range Prediction Results

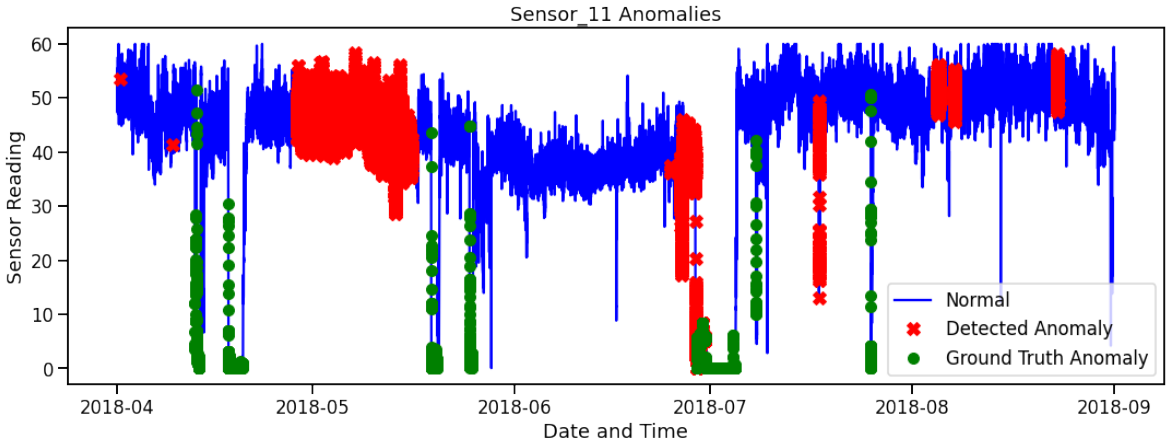

Figure 4.2: Sensor 11 Interquantile Range Prediction Results

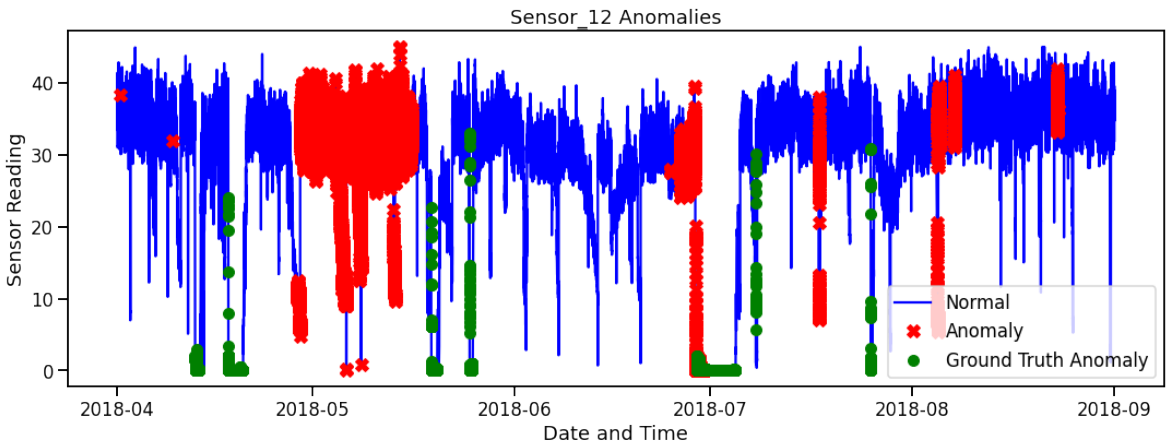

Figure 4.3: Sensor 12 Interquantile Range Prediction Results

4.5.2 Model 2 : K-means

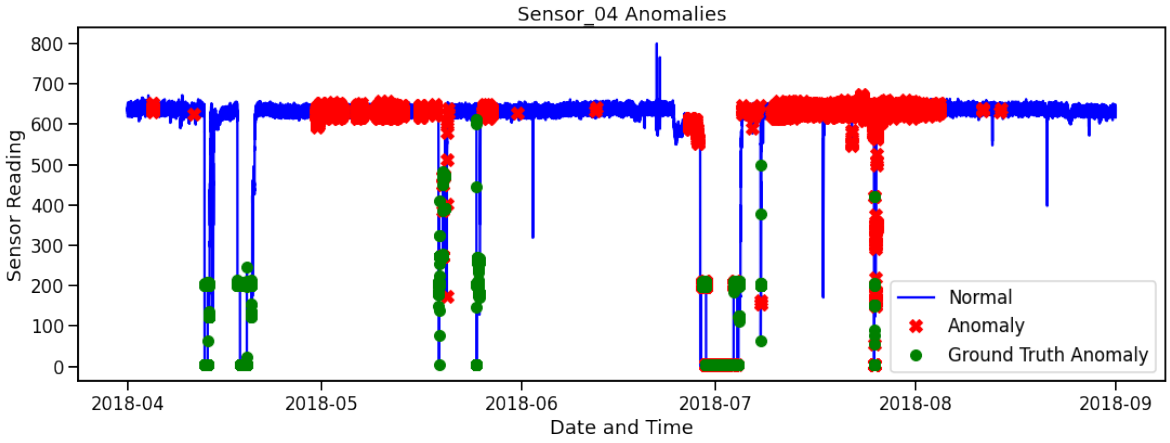

Figure 4.4: Sensor 04 K Means Prediction Results

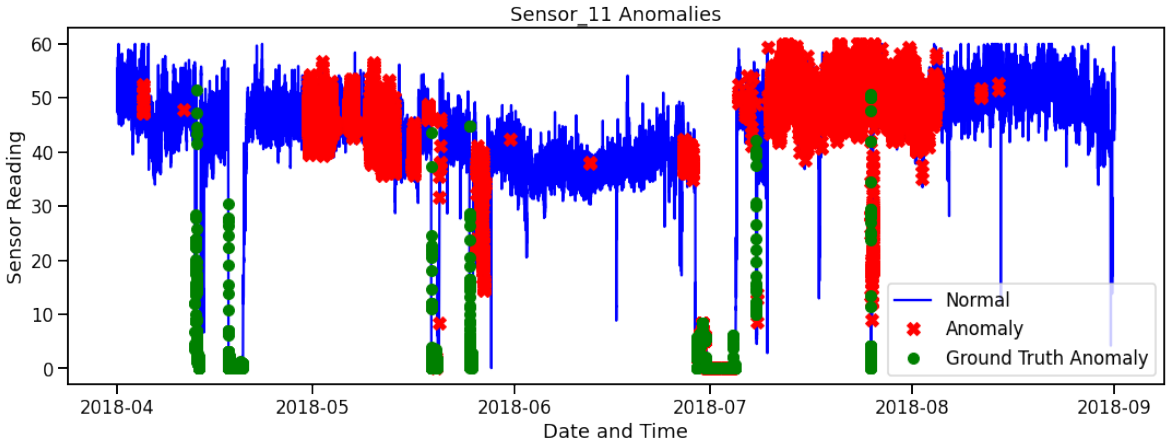

Figure 4.5: Sensor 11 K Means Prediction Results

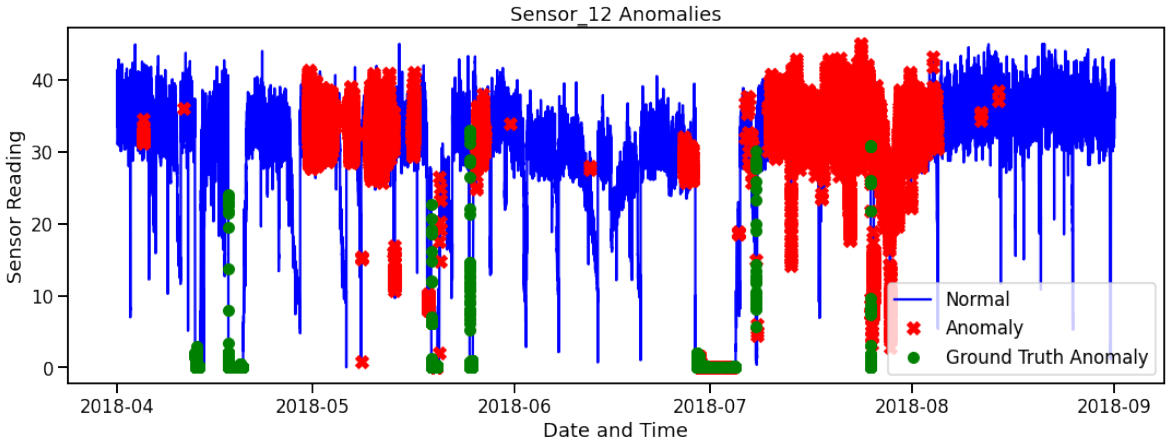

Figure 4.6: Sensor 12 K Means Prediction Results

4.5.3 Model 3 : Isolation Forest

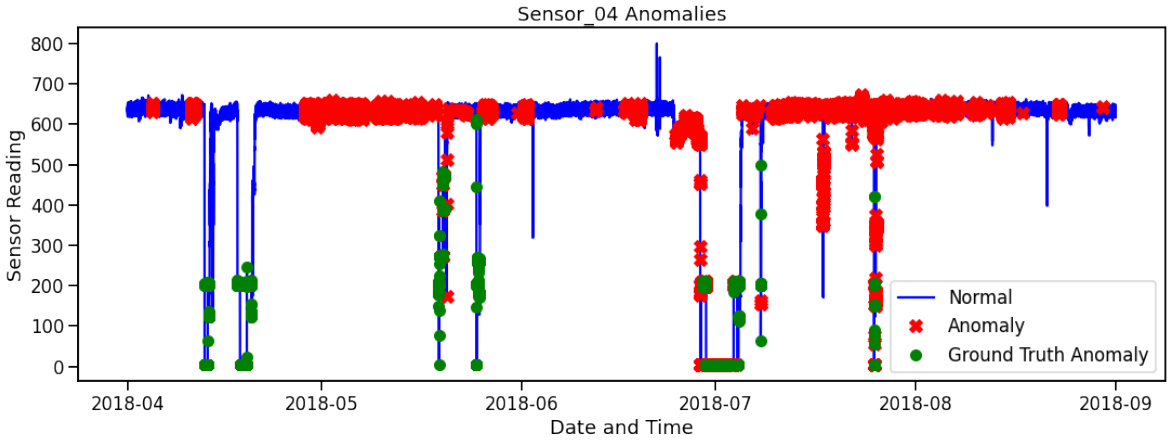

Figure 4.7: Sensor 04 Isolation Forest Prediction Results

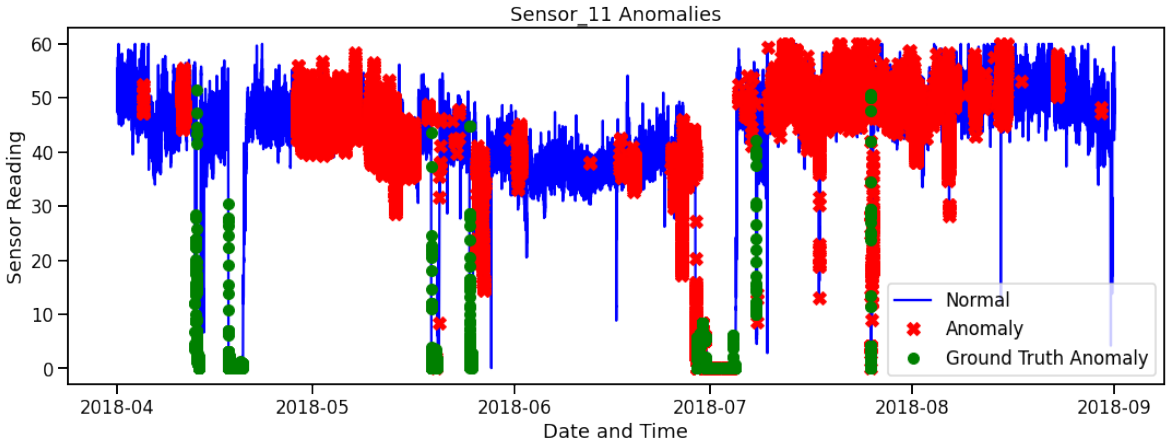

Figure 4.8: Sensor 11 Isolation Forest Prediction Results

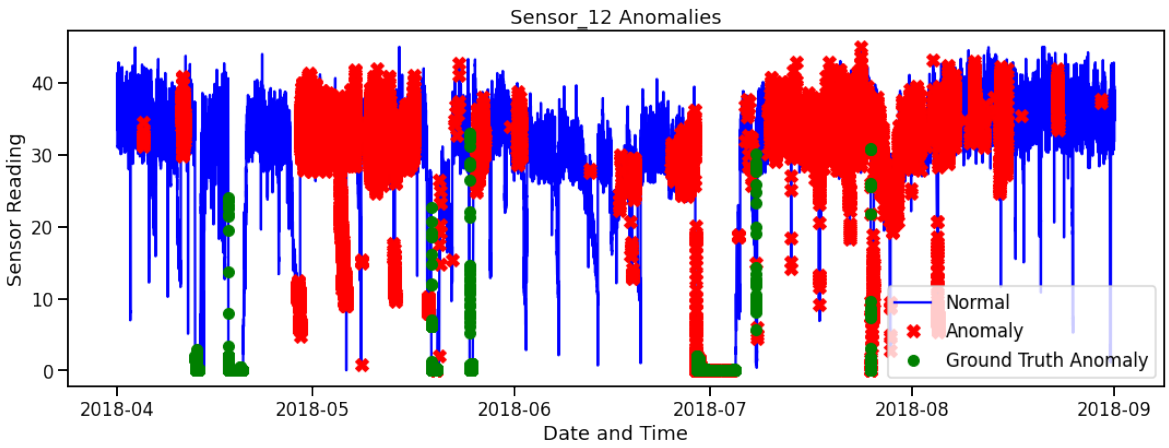

Figure 4.9: Sensor 12 Isolation Forest Prediction Results

4.5.4 Model 4 : One-Class SVM

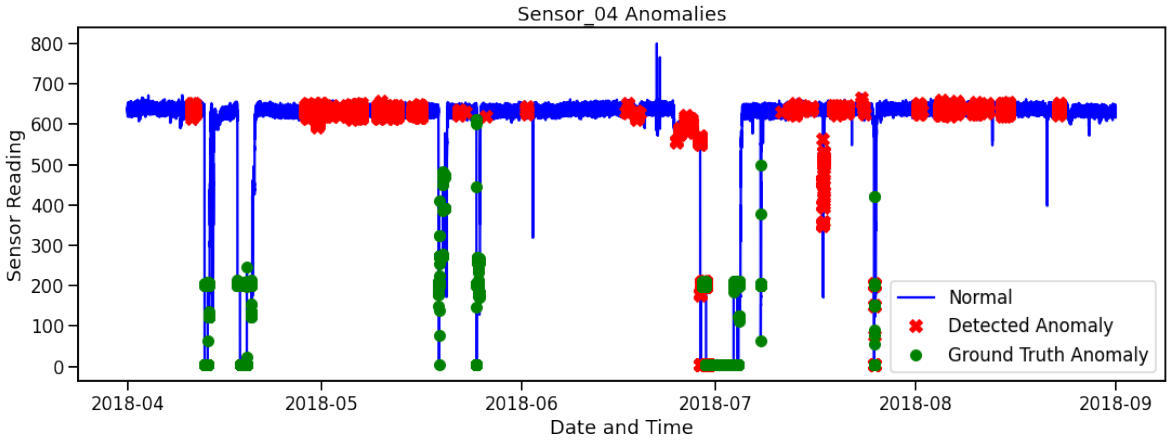

Figure 4.10: Sensor 04 One-Class SVM Prediction Results

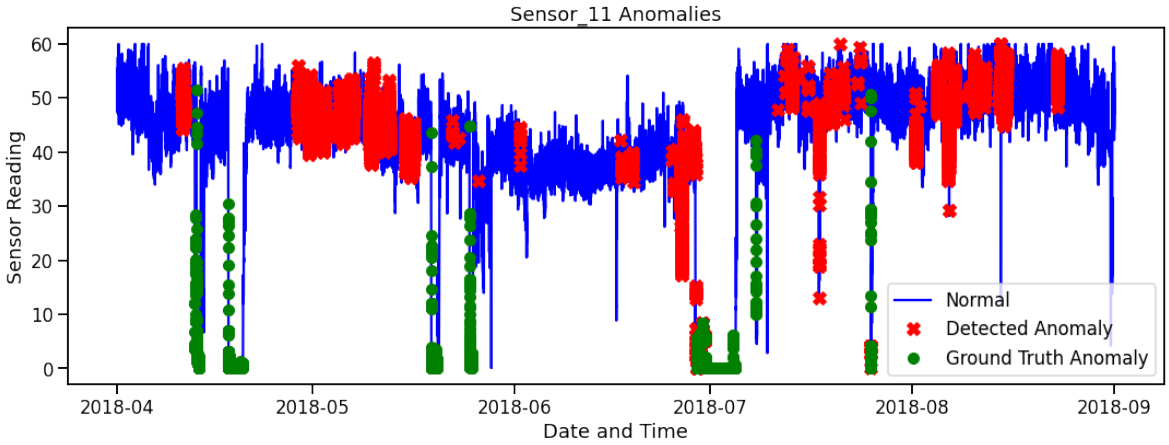

Figure 4.11: Sensor 11 One-Class SVM Prediction Results

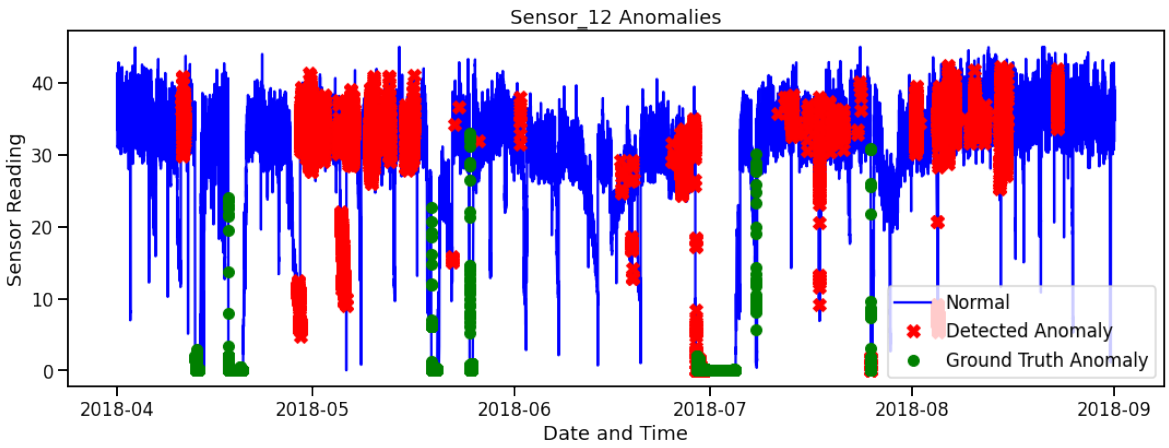

Figure 4.12: Sensor 12 One-Class SVM Prediction Results

4.5.5 Model 5 : Prophet

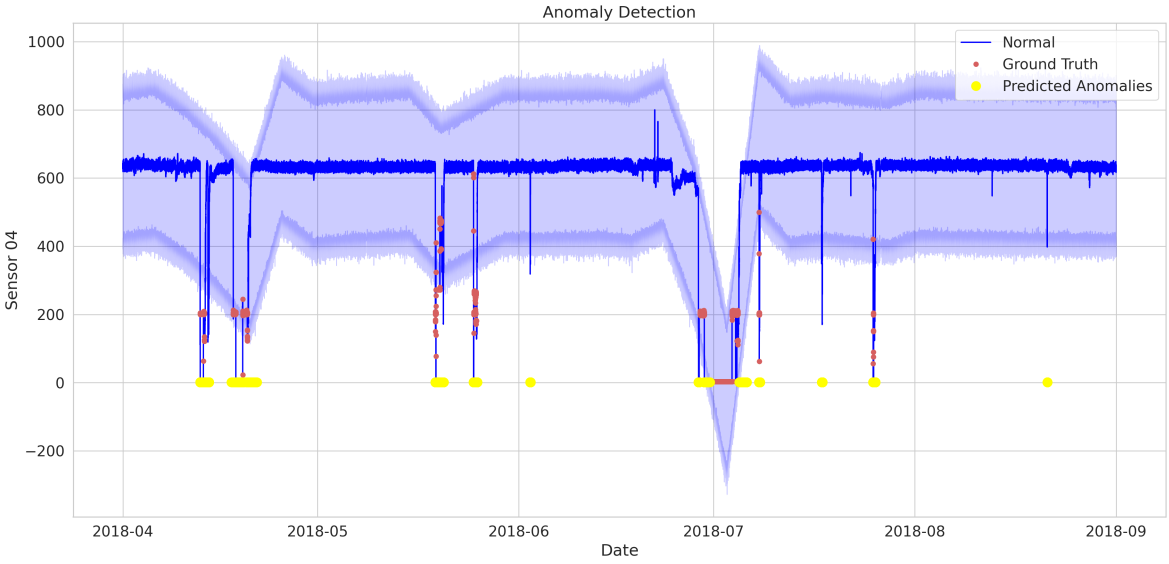

Figure 4.13: Sensor 04 Prophet Prediction Results

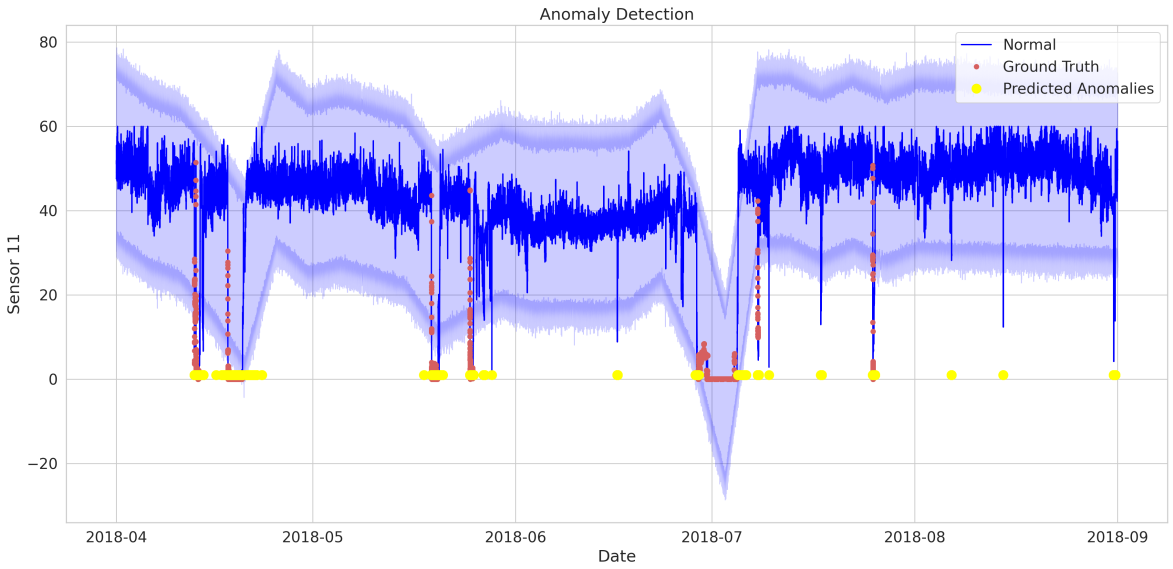

Figure 4.14: Sensor 11 Prophet Prediction Results

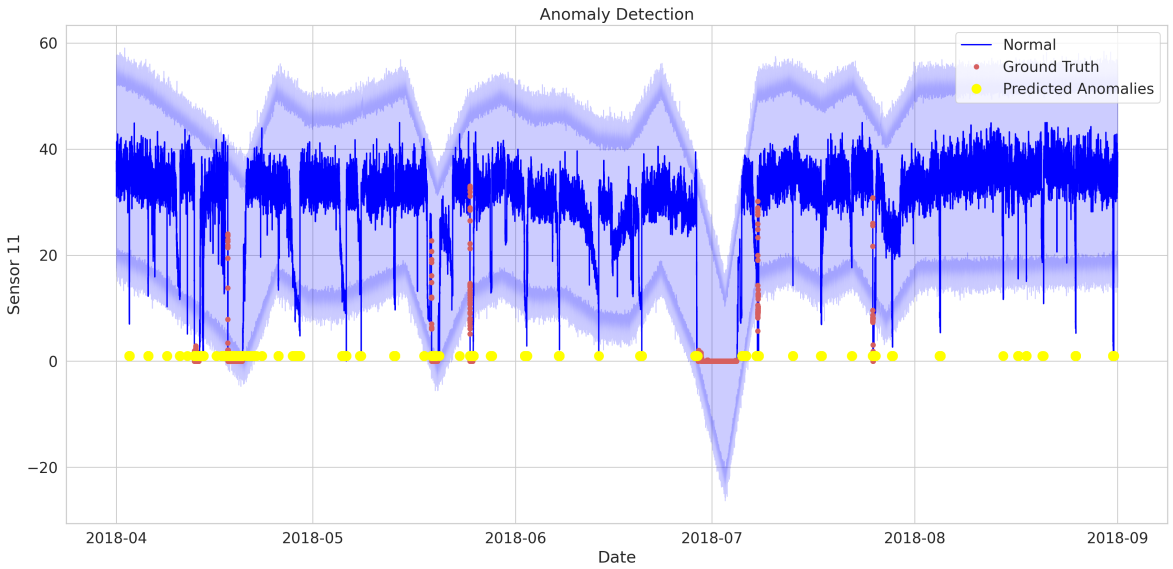

Figure 4.15: Sensor 12 Prophet Prediction Results

4.5.6 Model 6 : ADS - LSTM

In this project, we used LSTM models for anomaly detection on three sensors (sensor04, sensor11, and sensor12) in a manufacturing process. The goal was to identify anomalous behavior in the sensor readings, which may indicate potential issues in the manufacturing process.

First, we split the data into training and testing sets based on a threshold date of June 15, 2018 with specified timestep of 10 minutes. We then used a standard scaler to normalize the training data and created a sliding window approach with a time step of 10 minutes to prepare the data for LSTM modeling.

Next, we built an LSTM model for each sensor to predict the likelihood of anomalies in the sensor readings. The models consisted of three LSTM layers with dropout regularization followed by a dense layer with an activation function. We used class weights to handle the class imbalance in the training data and implemented early stopping to prevent overfitting.

After training the three models, we combined their results using Majority Vote Ensemble Method to generate a final prediction for each time step. An anomalous reading was classified as any time step where the majority of the three models predicted an anomaly.

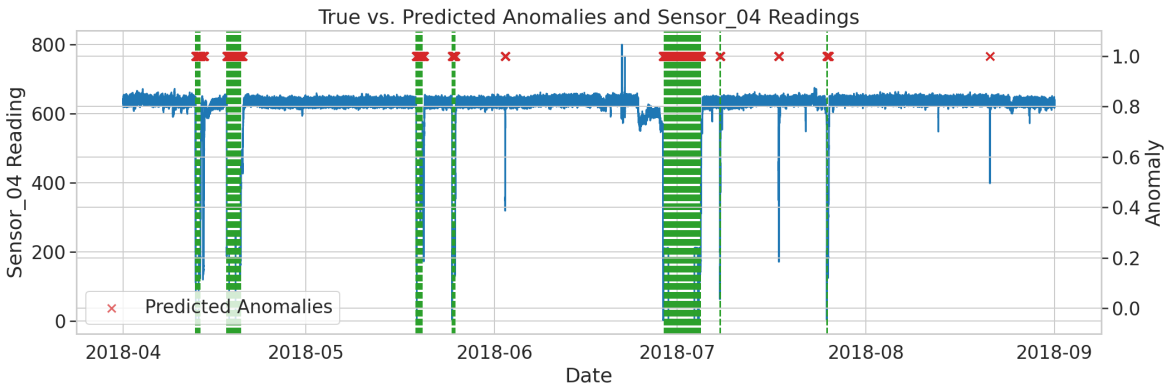

Figure 4.16: Sensor 04 LSTM Prediction Results

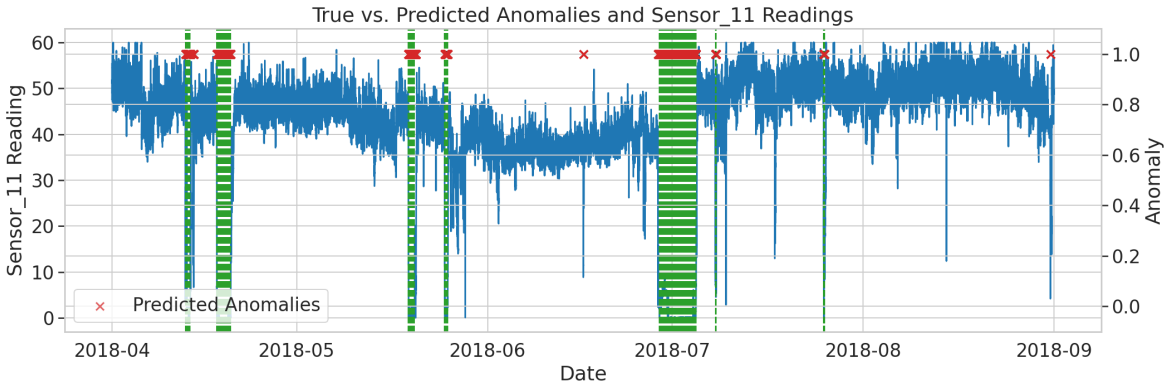

Figure 4.17: Sensor 11 LSTM Prediction Results

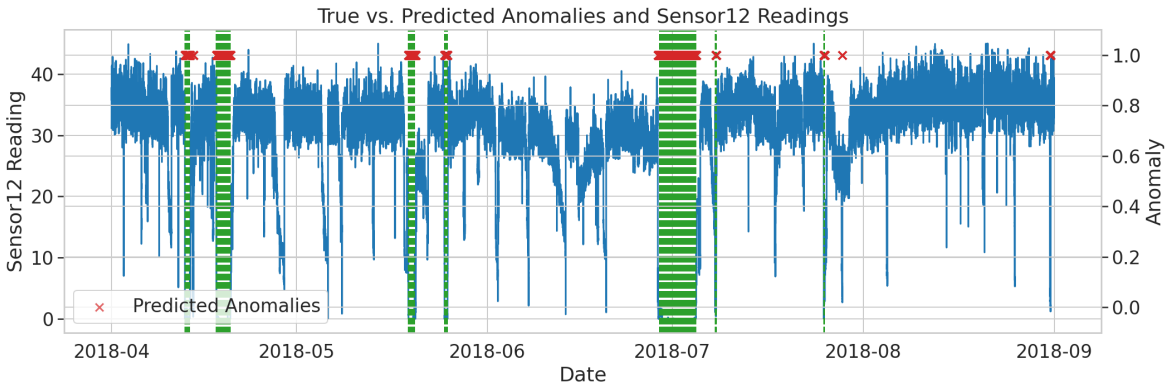

Figure 4.18: Sensor 12 LSTM Prediction Results

Chapter 5

Conclusion

In this study, we investigate the performance of various anomaly detection models for identifying anomalous events within a large-scale dataset. Our main objective is to determine the most effective model for this specific task, considering several evaluation metrics. We thoroughly analyzed the dataset, performed preprocessing steps, and implemented six different anomaly detection models: IQR, K-means, Isolation Forest, One Class SVM, Prophet, and ADS-LSTM.

Our evaluation takes into account multiple performance metrics, such as accuracy, precision, recall, F1-score, and AUC. The results indicated that the ADS-LSTM model outperforms the other methods across all metrics, showcasing its effectiveness and accuracy in detecting anomalies. Additionally, we utilize Matplotlib and Prophet to visualize the results, which provide deeper insights into the models' performances and enhanced our understanding of the detected anomalies.

Although our research has contributed valuable insights into the effectiveness of different anomaly detection models applied to this dataset, there are opportunities for improvement and further exploration. The following potential avenues for future research are suggested:

1. **Incorporate additional features:** Our current models rely on the given set of features from the dataset. Integrating additional features, such as external factors or engineered features, could potentially enhance the models' performance by providing more information for anomaly detection.

2. **Explore novel algorithms and techniques:** We focused on six different anomaly detection models in this study. However, the field of anomaly detection is continuously

evolving, with new techniques and algorithms being developed [34]. Future work could investigate the effectiveness of other state-of-the-art methods in detecting anomalies in the dataset [56].

3. **Hybrid models:** Combining the strengths of multiple models could lead to improved performance [57]. Future research could explore the development of hybrid models that leverage the best aspects of different anomaly detection techniques to create a more robust solution [**?**].

4. **Fine-tuning model parameters:** We used default or expert-suggested parameters for the models in our study. However, it is possible that some models could benefit from further parameter tuning to optimize their performance on the dataset [58]. Future work could involve performing a systematic exploration of the parameter space for each model to identify the best possible configurations [59].

5. **Transfer learning:** Transfer learning, which involves using pre-trained models or transferring knowledge from related tasks, could be a promising approach to improve the performance of anomaly detection models [60]. Investigating the potential for transfer learning in this domain would be a valuable contribution to the field [61].

6. **Real-world implementation:** Our research has focused on evaluating the models' performance on a specific dataset. However, the true test of an anomaly detection model lies in its ability to generalize to real-world settings [62]. Future work could involve deploying these models in real-life situations to assess their effectiveness and robustness in practical applications [63].

In conclusion, our study adds to the growing body of knowledge in the field of preventive maintenance by demonstrating the strengths and limitations of various models when applied to a large-scale time series dataset of sensors. Our findings lay a strong foundation for future research, which will further advance the development of more effective and reliable anomaly detection systems.

Bibliography

[1] John Smith et al. Industrial equipment monitoring using wireless sensor networks and anomaly detection. *IEEE Transactions on Industrial Informatics*, 14(4):1642–1651, 2018.

[2] Chen et al. Anomaly detection in industrial sensor data for preventive maintenance. *International Journal of Prognostics and Health Management*, 10(2):1–12, 2019.

[3] Yang et al. Time series anomaly detection for wind turbine monitoring. *Renewable Energy*, 89:671–679, 2016.

[4] Li et al. A time series anomaly detection method for power transformer equipment in smart grids. *Applied Energy*, 197:323–332, 2017.

[5] Nayak et al. Real-time sensor data analytics for railway transportation systems. *Transportation Research Part C: Emerging Technologies*, 104:314–331, 2019.

[6] Wang et al. Anomaly detection in aircraft flight data using a hybrid model. *Engineering Applications of Artificial Intelligence*, 45:220–233, 2015.

[7] Zhao et al. Anomaly detection of urban bridge health monitoring data using time series data mining. *Structural Health Monitoring*, 17(2):437–450, 2018.

[8] Kim et al. Anomaly detection in smart grid data: A time series approach. *Applied Energy*, 195:959–969, 2017.

[9] Alippi et al. A sensor network-based monitoring system for environmental monitoring. *Sensors and Actuators B: Chemical*, 146(1):139–147, 2010.

[10] De et al. Water quality monitoring using wireless sensor networks: An anomaly detection approach. *Water Resources Management*, 30(15):5693–5707, 2016.

[11] Mishra et al. Anomaly detection in wearable sensor data for monitoring patients with chronic diseases. *Journal of Medical Systems*, 43(9):1–14, 2019.

[12] Li et al. A survey on time series data mining for medical devices. *Biomedical Signal Processing and Control*, 17:68–78, 2015.

[13] Dey et al. Anomaly detection in iot sensor data using time series analysis. *Proceedings of the International Conference on Sensing, Signal Processing, and Security*, pages 165–173, 2017.

[14] Yun et al. A time series anomaly detection method for iot device using deep learning. *Sensors*, 18(4):1123, 2018.

[15] Brockwell et al. *Introduction to Time Series and Forecasting*. Springer, 2002.

[16] Hyndman et al. *Forecasting with exponential smoothing: the state space approach*. Springer, 2008.

[17] Box et al. *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition, 2015.

[18] Cleveland et al. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[19] Rabiner et al. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[20] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[21] Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[22] Jolliffe et al. *Principal Component Analysis*. Springer, 2 edition, 2002.

[23] Liu et al. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[24] Schölkopf et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[25] Breunig et al. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104. ACM, 2000.

[26] Cover et al. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[27] Hinton et al. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[28] Sakurada et al. Anomaly detection using autoencoders with nonlinear dimensionality reduction. *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4, 2014.

[29] Hochreiter et al. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[30] Malhotra et al. Lstm-based encoder-decoder for multi-sensor anomaly detection. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2016.

[31] Kingma et al. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

[32] An et al. Variational autoencoder based anomaly detection using reconstruction probability. In *SNU Data Mining Center Tech. Rep*, volume 2, 2015.

[33] Lazarevic et al. A comparative study of anomaly detection schemes in network intrusion detection. *Data mining and knowledge discovery*, 8(4):281–316, 2005.

[34] Chandola et al. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009.

[35] Hochenbaum et al. Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv:1704.07706*, 2017.

[36] Malhotra et al. Lstm-based encoder-decoder for multi-sensor anomaly detection. In *International Conference on Machine Learning*, pages 89–97, 2016.

[37] Zhang et al. A deep learning approach for unsupervised ensemble learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[38] Ahmed et al. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[39] Laptev et al. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1939–1947, 2015.

[40] Susto et al. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2014.

[41] Goldstein et al. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[42] Feng et al. Robust anomaly detection in sensor data with noise. *Sensors*, 17(12):2844, 2017.

[43] Kieu et al. Unsupervised deep learning for anomaly detection in cyber physical systems. *arXiv preprint arXiv:1809.05078*, 2018.

[44] Keogh et al. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM)*, page 8. IEEE, 2005.

[45] Lin et al. Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[46] Wei et al. A nearest-neighbor approach for feature selection in regression. In *Proceedings of the Fifth SIAM International Conference on Data Mining*, pages 204–215. SIAM, 2005.

[47] Zhu et al. Statstream: Statistical monitoring of thousands of data streams in real time. *The VLDB Journal*, 12(2):81–96, 2003.

[48] Goldin et al. A learning automata-based approach to temporal pattern discovery. In *Proceedings of the 2007 ACM symposium on Applied Computing*, pages 633–638. ACM, 2007.

[49] Gupta et al. Detecting heterogeneous changes in data streams. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 559–570. SIAM, 2010.

[50] Rakthanmanon et al. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM, 2012.

[51] Yeh et al. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322. IEEE, 2016.

[52] Senin et al. Grammarviz 3.0: Interactive discovery of variable-length time series patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(1):1–28, 2018.

[53] Wang et al. Anomaly detection in time series data using variable-length markov chains. *Machine Learning*, 102(3):273–304, 2013.

[54] Vahdatpour et al. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pages 1261–1266. AAAI Press, 2009.

[55] Chakrabarti et al. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 151–162. ACM, 2002.

[56] Hundman et al. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pages 387–395, 2018.

[57] Zhang et al. A hybrid unsupervised anomaly detection model for multivariate time series. *Knowledge-Based Systems*, 163:329–338, 2019.

[58] Bergstra et al. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[59] Snoek et al. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[60] Pan et al. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[61] Ravi et al. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1709.08707*, 2017.

[62] Gornitz et al. Toward supervised anomaly detection. *Journal of Machine Learning Research*, 16(1):773–826, 2015.

[63] Buczak et al. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, 2016.