Thesis for the Degree of Master of Engineering

A Study on the Data-Driven Approach for Anomaly Detection in Electric Power Steering System

by

Alabe Lawal Wale

Department of Artificial Intelligence Convergence

The Graduate School

Pukyong National University

August, 2023

Thesis for the Degree of Master of Engineering

A Study on the Data-Driven Approach for Anomaly Detection in Electric Power Steering System

전동식 파워 스티어링 시스템의 이상 징후 감지를 위한 데이터 기반 접근법에 관한 연구

Advisor: Prof. Youngsun Han

by

Alabe Lawal Wale

A thesis submitted in partial fulfillment of the requirements

For the degree of

Master of Engineering

Department of Artificial Intelligence Convergence

The Graduate School

Pukyong National University

August, 2023

A Study on the Data-Driven Approach for Anomaly Detection in Electric Power Steering System

A dissertation

by

Alabe Lawal Wale

Approved by:

Professor Won-Du Chang
(Chairman)

Professor Hoon-Hee Kim
(Member)

Professor Youngsun Han
(Member)

August 18, 2023

Acknowledgement

I thank my creator for His bountiful bestowal on me through my journey of life, starting from my earliest educational experiences to the accomplishment of my master's degree.

I would like to express my appreciation to my advisor Prof. Youngsun Han for his support over the last two years. Additionally, I use this opportunity to appreciate the National Institute for International Education, which acts as the coordinating body for the esteemed Global Korea Scholarship.

My deepest gratitude to my parents and beloved spouse for their unwavering support and affection. I love you all wholeheartedly. Lastly, I would like to acknowledge all my lab mates and friends for their support throughout my master’s program.

Alabe Lawal Wale

Department of Artificial Intelligence Convergence

Pukyong National University, Busan, South Korea

Contents

Figure List

Table List

A Study on the Data-Driven Approach for Anomaly Detection in Electric Power Steering System

Alabe Lawal Wale

Department of Artificial Intelligence Convergence, The Graduate School,
Pukyong National University

Abstract

With the centralization of anomaly detection in electrical power steering (EPS) systems through modeling and knowledge-based methodologies, the EPS systems have evolved to become intricate and advanced, necessitating heightened levels of quality assurance and general safety. Given that the majority of existing detection methods are reliant on pre-existing knowledge, accurately identifying novel or previously unobserved anomalies poses a challenge. In this study, a deep learning approach consisting of a dual-stage process involving an autoencoder and long short-term memory (LSTM) network for detecting anomalies within the data captured from the EPS sensor is presented.

The model was trained on EPS data utilizing an autoencoder to extract and compress features into a latent representation. Subsequently, the compressed features are inputted into an LSTM network to capture interdependencies among

the features and then reconstructed to obtain an output. An anomaly score was computed based on the reconstruction loss of the output, enabling the detection of anomalies.

The efficiency of the presented approach was substantiated through the collection and analysis of sample data obtained from an experiment conducted with an EPS test jig. Comparative results reveal that the proposed model outperforms other methods in anomaly detection, exhibiting an accuracy of 0.99 and a higher region under the receiver operating characteristic curve. This establishes the proposed approach as a robust tool for EPS anomaly detection.

전동식 파워 스티어링 시스템의 이상 징후 감지를 위한 데이터 기반 접근법에 관한 연구

알라비 라왈 왈레

부경대학교 대학원 인공지능융합 학과

초 록

모델링과 지식 기반 방법론을 통해 전동식 동력 조향(EPS) 시스템의 이상 감지를 중앙 집중화 함에 따라 EPS 시스템은 더욱 복잡하고 고급화된 형태로 발전하여 높은 수준의 품질 보증과 일반 안전이 요구된다. 기존의 대다수 감지 방법이 이전에 감지된 경험적 지식에 의존하고 있다는 점을 감안할 때, 새롭거나 이전에 관찰되지 않았던 이상 현상을 정확하게 식별하는 것은 해결해야 할 과제이다. 본 연구에서는 EPS 센서에서 측정된 데이터의 이상 현상을 감지하기 위해 오토인코더와 장단기 메모리(LSTM) 네트워크로 구성된 2 단계 프로세스의 딥러닝 접근법을 제시한다.

이 모델은 오토인코더를 사용하여 EPS 데이터를 학습시켜 데이터 특징을 추출하고 잠재 표현으로 압축한다. 그런 다음 압축된 특징을 LSTM 네트워크에 입력하여 특징 간의 상호 의존성을 파악하고 재구성하여 출력한다. 이상 점수는 출력의 재구성 손실을 기반으로 계산되어 이상 현상을 감지하는데 사용된다.

제안된 접근법의 효율성은 EPS 테스트 장치를 사용하여 수집된 샘플 데이터의 수집과 분석을 통해 입증되었다. 비교 결과, 제안된 모델은 이상 감지에서 0.99 의 정확도와 수신기 동작 특성 곡선 아래에서 더 높은 영역을 나타내어 다른 기존 방법보다 우수한 성능을 보였다. 이를 통해 제안된 방식이 EPS 이상 감지를 위한 강력한 도구임을 확인하였다.

CHAPTER ONE

1.1 Background and Motivation

Early steering systems were characterized by either a heavy or softly gearing mechanism. This attribute raises several difficulties within the automotive system, including augmented driver exertion, which refers to the increased physical effort required by the driver to control the steering wheel. Also, it introduces delays in response time, impeding the driver's capacity to execute accurate and finely tuned steering maneuvers, diminished stability, and reduced agility. To address these limitations, a power-assisted steering system was developed. EPS enhanced the vehicle control system significantly compared to the hydraulic power steering system as it required less steering effort, resulting in power efficiency since it is powered by the alternator rather than the engine (when the vehicle is powered) [1,2]. According to [3], the EPS market is expected to increase significantly from $23.03 billion in 2021 to $35.30 billion by 2029 with the continuous popularity of hybrid and self-driving vehicles. In addition, EPS offers numerous advantages to this vehicle design including its ability to lower the vehicle weight, conserve fuel consumption, and reduce the risk of vehicle intrusion through the usage of the Internet of Things (IoT) [4,5].

As the implementation of EPS becomes more prevalent in the automotive sector, there is a heightened demand for increased safety, reliability, and

performance [6]. These objectives can be achieved through the implementation of an optimized system design during the manufacturing phase, improved anomaly detection methods, and the integration of EPS health monitoring and prognostics techniques. The EPS system is composed of various mechanical and electrical components, including a torque sensor, handwheel angle sensor, vehicle speed sensor, and an electronic control unit. These components operate in a synchronized manner [7]. Potential EPS system malfunctions can range from anomalies or failures in individual components such as sensors and actuators, to impending failures such as insulation degradation of the stator coil or bearing issues that affect friction. Early diagnosis of the component fault, specifically those resulting in erroneous steering assist due to sensor malfunction is considered crucial as the outcome of such failure often results in disastrous outcomes due to driver shock [7]. Nevertheless, the risk mitigation strategies for EPS failure have not been thoroughly examined, leaving the potential for critical issues impacting the reliability for quality assurance and safety design of the EPS system.

Anomaly detection is a method used to identify abrupt deviations in system behavior from regular patterns. Such deviations usually arise from system failures that could be sudden or gradual in nature [8]. There are three primary methods for identifying these anomalies which are knowledge, model, and data-driven. Recently, the potential of the data-driven approach has increased

significantly due to the advancement in deep learning models and the advent of Industry 4.0. The emergence of Industry 4.0 has led to a transformative impact on the manufacturing sectors facilitated by the widespread use of Internet of Things (IoT) sensors that generate large-scale datasets. These types of datasets also known as time series data have been widely embraced for decision-making in the areas of anomaly detection [9], intrusion detection [10], and predictive maintenance [11]. However, the acquisition of a dataset that contains instances of anomalous events is a challenging and expensive task, which restricts the application of the conventional data-driven method. This limitation is now addressed with deep learning models using an unsupervised or semi-supervised learning method.

In this thesis, an anomaly detection framework is proposed for the EPS system leveraging the capabilities of deep learning algorithms to eradicate inferior components. The architecture comprises a hybrid model: A long short-term memory network, which is an extended version of the recurrent neural network, to retain temporary dependencies on each data point and an autoencoder to compress the high dimensional dataset into a latent space for calculating the reconstruction loss used as the anomaly detector.

1.2 Thesis Contribution

1. The prevalent approach for anomaly detection in EPS parts is knowledge and modeling-based approaches. In this thesis, a data-driven method using deep learning is proposed.
2. A two-stage approach is implemented in this proposed method. Firstly, normal data without anomalies is utilized for training the model and secondly, the anomaly detector is based on the reconstruction error using the mean absolute error.
3. The Dataset used for conducting the experiment is obtained from an EPS test jig and verifies the performance analysis to similar methods for anomaly detection using deep learning.

1.3 Thesis Outline

The thesis begins with the introduction and motivation of the research topic in chapter one. In chapter two, the theoretical foundation for understanding the thesis is discussed. Chapter three presents related works similar to this research theme. The methodology used in this research is described in chapter four, while chapter five presents the experimental results which comprise the dataset utilized, experimental setup, evaluation metrics, and performance analysis. Chapter six concludes the thesis and describes the direction for future works.

CHAPTER TWO

In this chapter, the theoretical knowledge required for understanding this thesis topic is introduced.

2.1 Machine Learning

Machine learning (ML) is a subset of artificial intelligence that enables computer systems to extract meaningful and potentially useful information without being explicitly programmed [12]. Artificial Intelligence (AI) is a field of study concerned with the development of intelligent agents- systems that can comprehend their environment and take action that enhances their chances of achieving their objective functions [13]. Essentially, AI encompasses the building of machines that are capable of performing tasks that are typically associated with human intelligence, such as problem-solving, and decision-making. Machine learning algorithms employ perceptron-based, statistical methods to detect patterns and meaningful insights in data and use these patterns to make predictions or decisions [14]. The core procedure of this machine-learning process can be categorized into three phases: training, testing, and application.

In the training phase, ML models are trained using training data depending on the learning algorithms. The aim of this phase is to optimize the effectiveness of the learning algorithm for the accurate prediction of unseen data [15]. The

validation and test stage involve evaluating the training algorithm performance on a separate set of data that was not utilized during the training stage. This ensures that the algorithm is not overfitting to the training data and will eventually generalize to unseen data. The application phase is the integration of the trained model for prediction or decision-making in a specific domain for real-world use. Figure 2.1 illustrates the different ML approaches discussed in this thesis and the required training data.

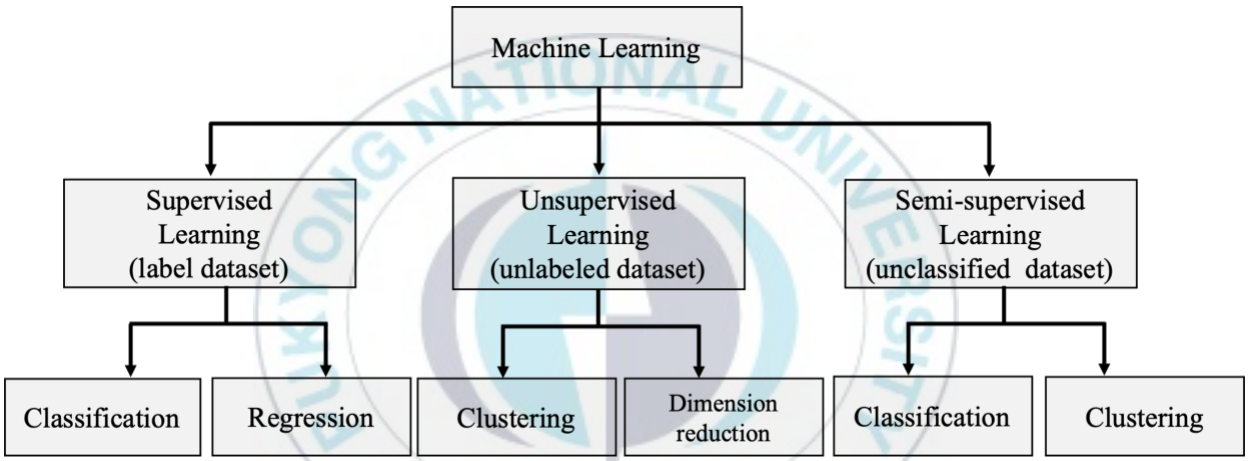


Figure 2.1: Different machine-learning approaches and data requirements.

2.1.1 Supervised Learning

Supervised learning is a fundamental task in machine learning that involves training an algorithm that maps an input to an output based on a sample-labeled dataset [16]. A supervised learning algorithm aims to create a model that can accurately predict the output for new input data. Classification and regression are the two most common tasks performed with supervised learning. The nature of the output value distinguishes classification and regression. In classification, the

output value is categorical or discrete, i.e., classify received emails as either spam or not spam, cloudy or rainy [17]. While in regression, the output value is continuous, i.e., prediction of housing price and stock rate. This distinction influences the choice of algorithms and evaluation metrics used for each task.

2.1.2 Unsupervised Learning

Unsupervised learning is a machine learning technique where the algorithm is trained to identify patterns and insights in the input data without being provided with explicit feedback or labeled samples [17-18]. The algorithm analyzes the input data to detect hidden structures and clusters, then uses this observation to create a model that can detect similar patterns in new data samples [18]. The aim of unsupervised learning is usually to explore and comprehend the underlying structure of the data rather than achieve a particular goal or prediction. Unsupervised learning is commonly used for clustering, i.e., splitting the samples in an untagged data collection into several typically separate subsets such as news stories and dimensionality reduction.

2.1.3 Semi-supervised Learning

Semi-supervised learning is a hybrid combination of both supervised and unsupervised learning approaches. Unlike the previous type of learning, where data samples are classified as either labeled or unlabeled, the state of data is not classified [19]. In the real-world use case labeled data are often limited, whereas unlabeled data is numerous, making semi-supervised learning effective. Hence,

the objective of semi-supervised learning algorithms is to deliver more effective prediction results than what can be achieved using only labeled data [19-20]. This learning method is widely applicable to fraud detection, machine translation, and anomaly detection where training instances are normal sample datasets.

2.2 Deep Learning

Figure 2.2 illustrates the connection and interdependences between Machine learning (ML) and Deep learning (DL) as a subfield of Artificial intelligence (AI).

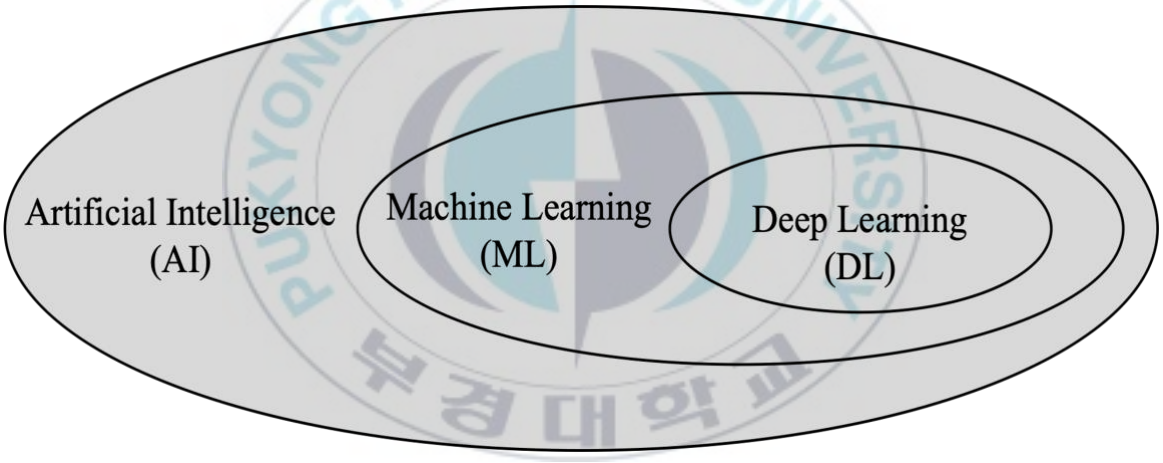


Figure 2.2: The relationship between AI, ML, and DL.

Deep learning is a subfield of machine learning that enables intelligence systems to leverage artificial neural networks to automatically extract, analyze and comprehend meaningful insights from a large dataset [21-22]. Unlike typical machine learning algorithms, deep learning models rely on multiple layers of interconnected neurons to automatically learn hierarchical data representation. Through the process of iteratively adjusting the neural network parameters, deep

learning models are able to identify and extract more abstract and relevant features from raw input data. Consequently, they have achieved remarkable success in tasks such as anomaly detection, image recognition, and natural language processing [22].

2.3 Anomaly Detection

Anomaly detection is the process of identifying data patterns that diverge from expected behavior. Depending on the context of the application, such deviant patterns are referred to as novelty detection, outliers, and discordant observations [23]. It is an intriguing field of machine learning research that entails the discovery of rare and fascinating patterns within datasets and has been widely applied in domain areas such as fraud detection, intrusion detection, cyber-attack, medical diagnosis, and fault detection in both electrical and mechanical systems [23]. Anomalies may occur in data due to a variety of factors, including deliberate actions such as fraudulent transactions with debit cards, malware attacks, or incidental events such as system breakdowns. Regardless of the manifestations, anomalies possess the attribute of being significant and remarkable to the investigator or analyst [23-24].

Generally, anomalies are categorized into point, contextual, and collective anomalies. Point anomaly occurs when a data instance significantly differs from the regular pattern of data [24]. A practical example can be observed when a house's daily power consumption is 10kwh but on a random day, and it abruptly

becomes 25kwh; this instance is considered a point anomaly. An anomaly within a particular context is termed a contextual or conditional anomaly. This type of anomaly is often characterized by contextual and behavioral attributes [23]. The most realistic scenario is household expenditure during the winter season compared to other seasons. The collective anomaly occurs due to changes in a group of interrelated data instances compared to the overall dataset. For instance, prolonged periods of low readings in human ECG indicate an exceptional phenomenon [23]. Figures 2.3 illustrate a generic framework for anomaly detection.

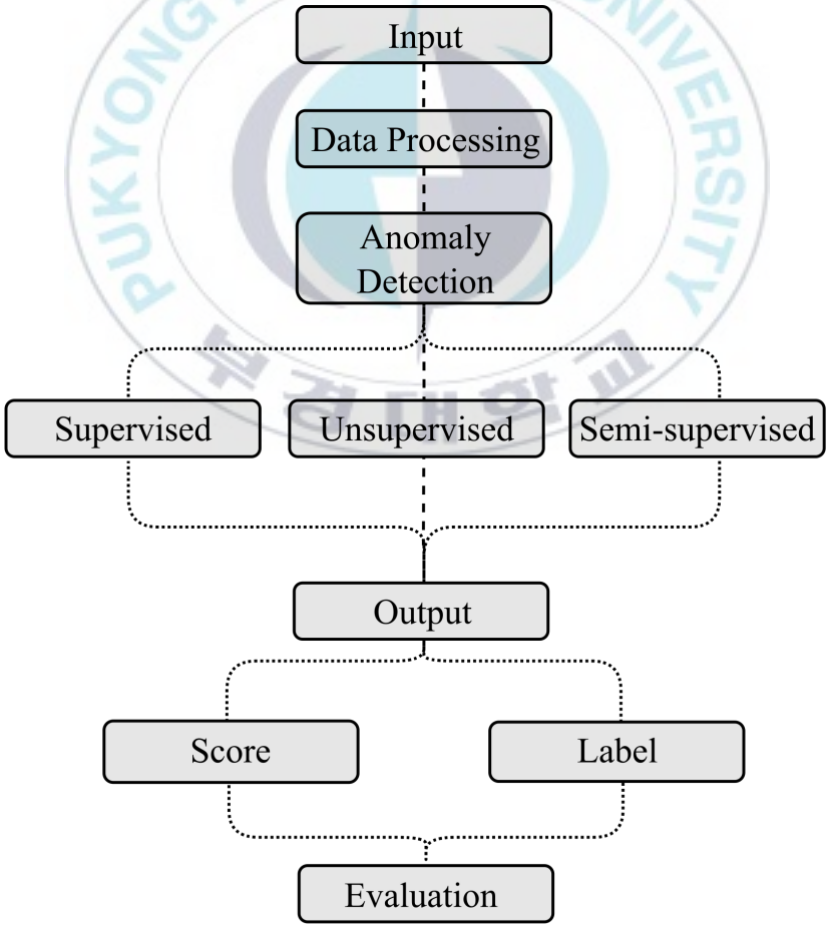


Figure 2.3: A generalized framework for detecting anomalies.

2.4 Working Principle of EPS

Several variants of Electric Power Steering (EPS) systems have been proposed, but they all operate based on a similar fundamental principle. As depicted in Figure 5.1, the driver exerts torque manually on the steering wheel. Based on the driver's input, the torque sensor measures the amount of force applied and subsequently conveys it to the electronic control unit (ECU) in the form of an electrical signal [25]. The ECU processes and compares the received signal with the steering assist force, which takes into consideration a variety of distinct elements such as steering angle and vehicle speed. The amplification of steering assistance torque received from the electric motor is facilitated by the reduction gear that is connected to the steering column. The aforementioned steering assistance torque is transmitted to the pinion gear, which is connected to the steering rack, in order to convert the rotary motion derived from the steering wheel into a lateral motion. This mechanism enables the steering rack to move the vehicle's wheels laterally, thus allowing the driver to steer the vehicle in the desired direction. Figure 2.4 presents the workflow of an EPS system.

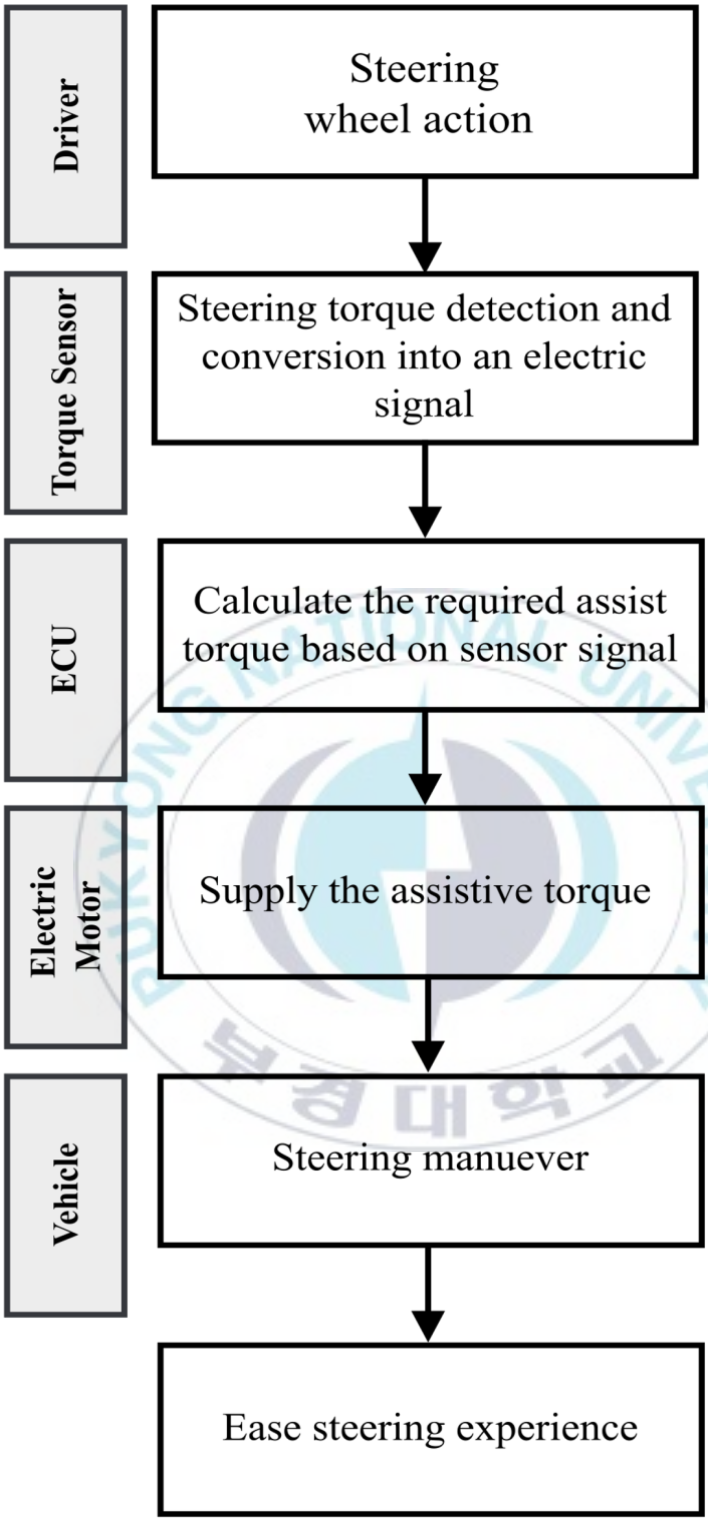


Figure 2.4: Workflow of EPS system.

CHAPTER THREE

This chapter summarizes previous work on anomaly detection related to this thesis. The techniques explored include conventional machine learning and deep learning approaches.

3.1 Traditional Machine Learning Approach

The efficiency of the classical machine learning approach for detecting anomalies is significantly influenced by the nature of the dataset being examined, as well as the training model employed. Depending on the specific application domain, these datasets may be labeled, partially labeled, or unlabeled. In instances where labeled data is accessible, a supervised learning method is employed for binary classification tasks to provide a distinct difference between normal and anomalous events.

The deployment of cloud computing has risen to prominence across both software service and corporate entities, owing to the significant decrease in both capital and operation expenses. This technological innovation enables an organization to leverage third-party service providers for computing resources such as storage, and networking infrastructure rather than deploying on-premises data centers. Hence, the broad adoption of cloud computing has been impeded by major security concerns against intrusion attacks. Tara et al. [26] present the use

of two classical machine learning models, linear regression, and random forest, to detect anomalies in a multi-cloud system. The authors studied the detection and classification of anomalies rather than solely focusing on detection. A publicly available is utilized to train and test the proposed approach, resulting in a commendable accuracy rate of 99 percent.

Assistive robotics holds immense potential for individuals with disabilities to carry out their daily tasks and activities independently. This concept is inspired by numerous instances, including but not limited to robot-supported feeding and clothing. Although robotic assistance provides substantial advantages in assisting people with daily chores, it is critical to ensure such systems' safe operation, efficiency, and reliability. This becomes extremely challenging when dealing with complex semi-autonomous systems that may occasionally experience anomalies, leading to potential safety risks and reduced system performance. Park et al. [27] proposed a technique to detect and classify anomalies in robot-assisted feeding. They employed a hidden Markov model for the anomaly detector, and a conditional log-likelihood classifier was implemented with selected input features obtained from the hidden Markov models. The proposed anomaly detector demonstrated a detection accuracy of 83.27% in identifying abnormal instances among 352 feeding trials, along with the ability to classify the kinds and underlying causes of the detected anomalies with 90% and 81% accuracy, respectively. However, the classifier utilized conditional log-likelihood

extracted from input signal sequences that fall below a limit that varies with time. This approach frequently faces the issue of class imbalance as the number of anomalies is significantly lower than the number of normal data in the training set [28]. Generating precise labels, particularly for the anomaly class, necessitates the involvement of domain specialists, which results in a tedious and costly process. Furthermore, scaling datasets with high dimensions is often challenging. Due to this shortcoming, unsupervised learning methods are often preferred.

Unsupervised learning methods exploit the spatial proximity of data points to identify anomalies, utilizing approaches such as density-based and distance techniques. A support vector machine (SVM) and density-based spatial clustering application with noise (DBSCAN) is proposed by Emadi et al. [29] to identify anomalies in wireless sensor networks using the IRLB dataset, which comprises eight features. In this study, three features were selected (temperature, humidity, and voltage) for their high impact on wireless network anomaly detection. Prior to using DSCAN, it is essential to assess the accuracy of the input features. The author hypothesized that two clusters would be required for the detection of anomalies based on density. Hence, the correlation coefficient between the input and output variables is determined prior to classifying high-density clusters as normal. On completion of the data labeling, the training is performed using a support vector machine.

Detecting anomalies or outliers in streaming data poses significant challenges compared to static data since data streams are characterized by continuous, dynamic, and unbounded changes. Mistra et al. [30] examined the difficulties of detecting outliers in data streams as well as the results of various outlier detection methods. The research focused on analyzing density-based techniques for detecting outliers and conducted a detailed comparison of various local outlier factor (LOF) based algorithms. they attributed a numerical value, called an outlier factor, to each data instance. The LOF metric is assigned a value of 1 to data points that are highly integrated into a cluster, while data points that are not well-integrated have a LOF value greater than 1. The author conducted a comparison of several LOF models and found that the memory-efficient incremental LOF algorithm is the most effective and scalable method for detecting anomalies in data streams.

Liu et al. [31] present a hybrid approach for detecting anomalies in large system logs that combines the K-prototype clustering technique with the K-nearest neighbor (KNN) classification algorithm. The K-prototype clustering algorithm uses both categorical and numerical data in the clustering phase, whereas the KNN classification model detects outliers by considering the distance of each log entry to its k-nearest neighbors in the dataset. The dataset is partitioned using k-prototype clustering into multiple clusters, based on the

extracted features attribute to precisely define usage patterns. Following the removal of the normal instances, which were typically evident as highly cohesive clusters, the remaining events were identified as outliers and necessitated further analysis. Consequently, the authors introduced two distance-based attributes to measure local and global outlier levels and employed K-nearest neighbor classifiers to assess the effectiveness of their method. The application of the K-mean clustering algorithm for network intrusion or anomaly detection is presented by Munz et al. [32]. The phrase "data mining" refers to a set of techniques and algorithms that aid the exploration of data with the aim to discover patterns and rules that can describe the intrinsic characteristic of the data.

However, the all-encompassing nature of data mining has limitations as the insights obtained may not be significant or practical in real-world scenarios. K-means clustering is a method that involves segregating features into K-distinct groups based on their unique attribute values. The grouping approach is designed to ensure that each group differs and is non-overlapping [33]. The training dataset was segmented into distinct clusters of temporals representing normal and anomalous traffic in their study. Subsequently, the corresponding cluster centroids are employed as templates for distance-based detection of anomalies in the investigated dataset. This method involves calculating the distance between the monitoring data and the established centroids, allowing anomalous patterns to be identified. An instance is considered an anomaly if its proximity to the

anomalous cluster centroid is closer than its proximity to the normal centroid or if its distance to the normal cluster centroid exceeds the predetermined threshold score.

In several fields of study, the prevalence of large-scale datasets has grown significantly. To effectively analyze and comprehend such datasets, there is a need for methods that can substantially reduce their dimensionality in a comprehensible manner while retaining most of the information contained therein. Numerous methodologies have been developed to achieve this objective, but Principal Component Analysis (PCA) is one of the earliest and perhaps widely adopted methods. The underlying principle of PCA is relatively straightforward: it involves the reduction of the dataset's dimensionality while retaining the maximum amount of variability within the dataset [34]. Kudo et al. [35] present a resilient principal component analysis (PCA) technique designed to identify anomalies in data traffic that exhibit daily or weekly periodic patterns. While PCA is an efficient technique for identifying traffic anomalies, it may be prone to contamination of the normal subspace when dealing with extremely large anomalies. This contamination could potentially degrade the effectiveness of the anomaly detector. In the proposed approach, the author leverages the cyclic characteristics of network traffic to detect anomalies at fixed intervals (e.g., daily). At the commencement of each period, a reference covariance matrix is formulated using normal traffic data from the preceding period. This reference

matrix serves as a benchmark for identifying deviations in the subsequent period's network traffic. Prior to employing PCA, outliers in the present period are eliminated by assessing them against the reference covariance matrix using the Mahalanobis distance technique. The presented approach effectively mitigated the issues of subspace leakage and a high false negative rate in detecting anomalies.

The efficiency of real-time mobile health applications is heavily reliant on the accuracy of sensor readings, which is imperative for delivering exceptional health care. Hence, concurrent sensor readings may be erroneous and lead to anomalous physiological measurements resulting from internal and external influences. As a result, anomalous readings have a noteworthy impact on the dependability of these applications and consequently impact the patient's well-being. Given the immense volume of data, novel data-driven approaches that incorporate dimensionality reduction techniques are necessary for detecting anomalous measurements and delivering reliable, personalized medical services. Lamia [36] et al. proposed an anomaly detection method for medical wireless body area networks. This approach utilizes PCA to analyze biomedical signals collected from sensors and detect abnormalities based on the prediction squared error. The proposed method integrates two crucial features: firstly, it integrates a sophisticated dimension reduction algorithm that utilizes the temporal and spatial correlations between observed vital signs with a multivariant anomaly

detection technique. This differs from classical PCA which is insusceptible to anomalies and eliminates the need for training with a reliable and annotated dataset, resulting in a significant operational advantage. Secondly, both processes employ an unsupervised approach that is robust and lightweight. Their experimental evaluation on a real-world medical dataset shows a high sensitivity rate (recall) and a low rate of incorrect identifications (false positives).

3.2 Deep Learning Approach

Deep learning has been utilized for resolving anomaly detection challenges through its advanced neural network architectures. Yunli et al. [37] present an innovative approach leveraging deep learning methodologies to enable efficient health monitoring of both heating and cooling equipment, thereby enhancing the overall reliability and performance of such systems. Condition monitoring is an integral component of system health management, serving as the primary stage in the process of defect detection, assessment, and prognosis. Thus, it plays a pivotal role in assuring the optimal operation and performance of complex systems, especially in the context of predictive maintenance and quality assurance. The process of conditional monitoring typically entails the observation and analysis of time series data captured by sensors, which commonly exhibit abnormalities such as outliers and transition instances that require close detailed examination and identification to enable effective monitoring and maintenance of the system. The approach adopted by the authors comprised a two-stage

process, wherein the first stage involved data prediction via an LSTM network, followed by anomaly detection through the utilization of the exponential weighted moving average (EWMA), leveraging the prediction errors generated by the LSTM model to facilitate accurate and efficient identification of anomalies in the data. The comparative analysis of performance demonstrates the superior efficacy of the LSTM algorithm over the alternative technique in accurately predicting and generalizing the detection of anomalies. Rui et al. [38] present a deep-learning method for machine health monitoring using the LSTM network. To conduct a robust empirical validation of the effectiveness of LSTMs, an experimental study was undertaken involving the operation of a high-speed computer numerical control machine in a dry milling environment. This enabled the collection of comprehensive data that served as a basis for examining the performance of the methodology under practical operating conditions, thereby enhancing the reliability and applicability of the findings. The approach adopted by the author demonstrated notable success in analyzing raw sensor data, utilizing both shallow and deep long short-term memory networks. The experimental findings provided compelling evidence that LSTM networks are capable of generating meaningful representations from raw sensor signals without the need for extensive feature engineering, thereby outperforming other deep learning models evaluated.

Kim et al. [39] introduced a novel method for web traffic anomaly detection that leverages the ability of convolutional neural networks (CNN) and the LSTM model, resulting in a unique convolutional long short-term memory (C-LSTM) framework. This cutting-edge approach enables the effective modeling of both spatial and temporal information within the traffic data, which is crucial for maintaining optimal system performance and ensuring a secure and reliable network infrastructure. In order to effectively capture the temporal information of the data, the LSTM network is used, while the CNN is utilized to mitigate the impact of spatial information frequency variation. Furthermore, to enhance the discriminative capacity of the model, a Deep Neural Network (DNN) is employed to map the input data into a more distinct feature space. Through a series of parametric trials, comparative model analyses, and comprehensive data evaluations, the authors identified the optimal model for their proposed method compared to existing machine learning models. As the proposed model employs a sliding window technique for data pre-processing, there is a delay in detecting anomalies in real-time data streams.

Although existing models for unsupervised anomaly detection that utilize dimensionality reduction followed by density estimation have shown notable advancements, they often encounter challenges such as disjointed model learning, irregular optimization objectives, and inadequate retention of critical information in the lower-dimensional space. Zong et al. [40] presented a novel approach to

address this issue by utilizing a deep autoencoding Gaussian mixture model for unsupervised anomaly detection. The proposed technique incorporates a deep autoencoder to create a low-dimensional entity and reconstruction error for each input instance, which is subsequently injected into the Gaussian combined model for anomaly detection. This method has the advantage of allowing the deep autoencoder to learn a robust feature representation that captures the essential data characteristics while eliminating the effects of noise. Malhotra et al. [41] introduced the long short-term memory autoencoder model (LSTM-AE) as a potential solution for detecting outliers in mechanical devices, utilizing the reconstructed output of sensor data. This method employs LSTM to develop an autoencoder model that extracts significant features from time-series data, which is then used to produce the reconstructed output for anomaly detection. Four real-world datasets were used to train the model: power demand, ECG, engine, and valve. According to the maximum likelihood estimation, it is hypothesized that any anomaly will result in a higher reconstruction output. The experimental outcome demonstrates that autoencoder is a robust technique capable of detecting anomalies in various types of time-series datasets, including regular, irregular, cyclical, non-cyclical, and semi-cyclical trends.

The swift development of the Industrial Internet of Things has brought a transformation in the traditional electrical grid, leading it towards a new digital era known as the Smart Grid. This paradigm shift has resulted in numerous

advantages, including optimized usage of the existing system, widespread control, and autonomous healing capabilities. Despite the advantages of smart grid technology, the proliferation of digital solutions has resulted in significant cybersecurity challenges, given the need to integrate with exiting vulnerable systems such as industrial control systems and supervisory control and data acquisition systems (SCADA). Ilas et al. [42] introduced a mechanism for detecting network-based threats and anomalies called MENSA that utilizes an innovative deep-learning framework combining autoencoders and generative adversarial network models. The proposed model aims to identify and classify operational anomalies in the system, such as those leveraging the Modbus/TCP protocol and Distributed Network Protocol 3. A generative adversarial network [43] is composed of two networks, namely the generator and the discriminator. The generator is responsible for creating synthetic data from random noise that mimics the distribution of real data. Meanwhile, the discriminator evaluates the generated data and determines whether it is genuine or fabricated.

The importance of Structural Health Monitoring (SHM) in maintaining the safety and sustainability of infrastructure, such as buildings and bridges, is escalating at a rapid rate. To this end, Son et.al [44] proposed a deep learning-based approach to identify outliers and classify incorrect data in a cable-stayed bridge. Cable-stayed bridges are prone to damage and potential collapse due to diverse factors such as environmental conditions, vehicular weights, and

corrosion of materials. The stayed cables in these structures are a critical component that significantly impacts structural soundness by transferring the bridge deck load to the supporting towers. Structural deterioration resulting from damaged cables can lead to a decrease in load-bearing capacity [45]. They employed a two-step method for detecting anomalies in a cable-stayed bridge, which involves utilizing an LSTM-autoencoder algorithm. In the initial stage, the anomalies were classified into two categories, namely persistent anomalies caused by structural damage and transient anomalies resulting from inaccurate data. The cables that contribute to the transient abnormalities are identified, and the abnormal data points are replaced with interpolation instances. In the second stage, the authors assessed the interpolated values for accuracy and suitability to determine if they were correctly classified and replaced with appropriate values.

CHAPTER FOUR

This chapter presents the overall structure of the proposed approach, with a specific focus on detecting EPS anomalies using an LSTM autoencoder. Previous studies have concentrated primarily on fault diagnosis, tolerance, and control of the EPS system using model-based and knowledge-driven methodologies. However, employing deep learning, particularly LSTM-AE, offers a distinct advantage by leveraging a data-driven approach. This method proves advantageous as it enables the detection of subtle abnormalities and progressive system degradation, which overcomes the constraints of traditional modeling-based approaches.

Figure 4.1 depicts the architecture of the proposed framework for identifying anomalies in EPS data. We employ a deep learning model on torque sensor time-series data. Torque sensors are often responsible for common sensor anomalies in EPS, which can result in difficulty turning the steering wheel, uneven power steering assist on the left and right sides, and decreased levels of assistive torque while driving [46]. Prior to the data being input into the LSTM-autoencoder model, the input data is preprocessed, and the model is trained using preprocessed data that solely consists of normal data from the obtained datasets of an EPS test jig. Finally, to detect anomalies in EPS data, the model's reconstruction errors are evaluated using test samples that include both normal and anomalous data.

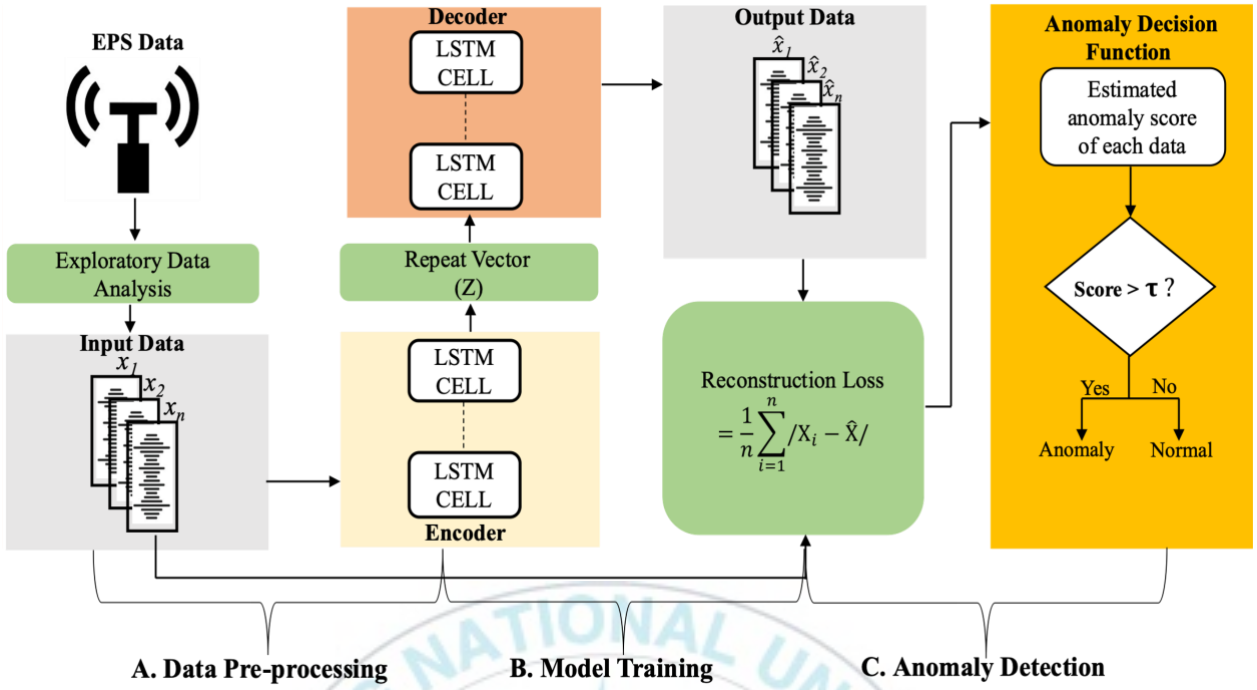


Figure 4.1: Layout of our proposed method for anomaly detection in EPS.

4.1 Pre-processing of Data

The first step in the implementation of deep learning models typically involves analyzing datasets to uncover initial patterns, identifying missing or incorrect values, and standardizing or normalization of data in preparation for the application of anomaly detection algorithms. Adopting such an approach is imperative as it helps to avoid a situation where specific data samples disproportionately affect the results, especially in algorithms that calculate feature distances, particularly when the datasets do not conform to a normal distribution. Such a skewed dataset can lead to suboptimal model performance, which highlights the necessity of preparing the data in a careful and systematic manner before it is used for training or testing purposes [47,48]. To avert such a

situation, the proposed framework employs normalization techniques to scale the dataset. Specifically, the sci-kit-learn MinMaxScaler is used to normalize the data between the range of 0 to 1. This scaling process is applied consistently to the training, validation, and testing datasets to ensure that the actual values are not altered during normalization. The equation for normalization is given below:

$$Z_i = \frac{x_i - min_{xi}}{max_{xi} - min_{xi}} \quad (1)$$

In the above equation, Z_i refers to the normalized value while x_i represents an individual data point from the original dataset.

4.2 Model Training

The training methodology entails utilizing a hybrid model that comprises both LSTM and autoencoder components.

4.2.1 LSTM Network

The LSTM network represents an advanced variant of the RNN architecture incorporating a long-term memory cell. In time-series data, the LSTM network supports the regulation of information flow, retains long-term dependencies, and identifies temporal correlations. This enables the LSTM network to perform significantly better than traditional RNN models with short-term memory [49], especially in scenarios where the data exhibits long-term

dependencies or recurring patterns. There exist several types of LSTM networks, with the two most prevalent variants being the vanilla and peephole architectures. The fundamental distinction between peephole and vanilla networks is in the connectivity across gates and memory cells. The cell state is updated in vanilla LSTM based on the input, output, and forget gates, which are determined by the current input, previous output, and prior cell state. The gates, however, do not have immediate access to the cell state. By contrast, Peephole LSTM allows gates to directly access the state of the cells through added connections. This enables gate values to be determined not only by the current input and prior output but also by the past cell state. Peephole LSTM networks have not been extensively utilized in research, despite their potential benefits. This is due to the divergent results that have been reported in recent studies, which have produced contradictory conclusions regarding the effectiveness of peephole networks in diverse applications [50,51].

Also, peephole LSTMs are computationally expensive to train and assess due to the additional parameters that result from the additional linkages between the cell state and the gates. Consequently, while peepholes tend to be an active area of research and development, researchers and practitioners continue to rely on vanilla LSTM networks for the vast majority of application cases. The introduction of working memory connections (WMC) is intended to enable accurate control of the gates and to mitigate the inconsistency associated with

peephole connection training [51]. Although the proposed model incorporating WMC has shown some promise, its efficiency appears to be limited, as there was only a substantial performance improvement observed during the training of the stacked LSTM model. In this proposed framework, I have opted to implement the widely used vanilla network, as detailed in [50]. The vanilla LSTM architecture, as depicted in Figure 4.2, consists of a memory cell, an input gate, an output gate, and a forget gate. The memory cell preserves information for arbitrary periods, while the three gates control the flow of information across the cell. Each gate in the vanilla LSTM model is constructed using a sigmoid layer that is preceded by a point-wise multiplication computation. The output of the sigmoid layer ranges between one and zero, this determines which instance from the prior hidden state and new input data are permitted to flow into the network.

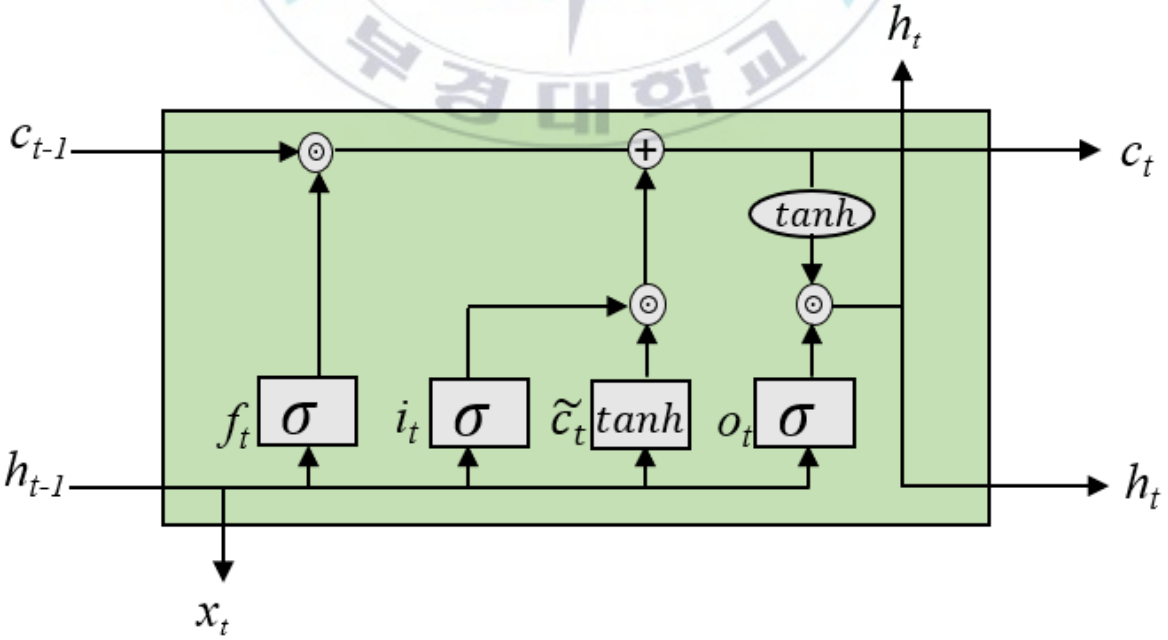


Figure 4.2: Schematic illustration of the LSTM architecture.

When an input vector x_t is introduced to the neural network at time t, the forget gate is responsible for regulating the information flow within the cell state, taking into consideration the previous hidden state and new input data. In general, the gate decides the relevance of the information that should be retained and the information that should be discarded. The relevant component is amplified and output as 1, while any unnecessary component of the input data is suppressed and output as 0. The forget gate's selective operation can be mathematically represented as follows:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (2)$$

In the given context, the activation function, denoted by σ, is associated with the weight and bias of the forget gate, represented by w_f and b_f, respectively. The primary function of the input gate can be attributed to two major objectives. Firstly, it is responsible for validating the significance of new information, which includes both the previous hidden state and the current input, with respect to the updated cell state. Secondly, it is responsible for updating the memory cell with the help of the aforementioned information. The input gate successfully accomplished these objectives by implementing a two-stage process. This process is similar to the forget gate; the initial step involves the determination of the information that holds relevance in the present memory cell. This procedure is described mathematically as follows:

$$i_t = \sigma(w_i [h_{t-1}, x_t] + b_i) \quad (3)$$

The activation function, denoted by σ, is accompanied by the weight and bias of the input gate, represented by w_i and b_i, respectively. The subsequent step involves the creation of a memory cell, denoted as $\tilde{c}_t$, which is accomplished by integrating the previous hidden state with the present input data. This operation employs a *tanh* activation function to create the memory update vector, the constituent values of which are confined within the interval of [1, -1]. The memory cell is characterized by the following definition:

$$\tilde{c}_t = tanh(w_c [h_{t-1}, x_t] + b_c) \quad (4)$$

The weight matrices and bias of the memory cell are represented in the above equation by w_c and b_c, respectively. To update the prior cell state c_{t-1}, Equation (2) and (3) are multiplied pointwise. The pointwise operation enables the network to selectively update cell state by blending previous cell state and the candidate cell state. The new cell state c_t is therefore obtained by adding the result of $f_t \odot c_{t-1}$ and $i_t \odot \tilde{c}_t$. This process is defined as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$

The output gate which is the last gate is computed as shown in Equation (6). The primary function is to determine the new hidden state by incorporating the current input data, the preceding hidden state, and the new memory cell. This is achieved by subjecting the old hidden state and the current input data to a sigmoid function.

$$o_t = \sigma(w_o\,[h_{t-1}, x_t] + b_o\,) \tag{6}$$

where, w_o and b_o represent the weight and bias of the output gate, respectively. The new hidden state is obtained by performing a pointwise multiplication of the filter vector o_t and the updated memory cell state c_t after it passes through the *tanh* activation function.

$$h_t = o_t \odot tanh(c_t) \tag{7}$$

The updated hidden state h_t and current memory cell c_t become the prior hidden state h_{t-1} and previous memory cell in the subsequent LSTM network. This iterative procedure continues until the whole sequence of input data has been processed.

4.2.2 Autoencoder

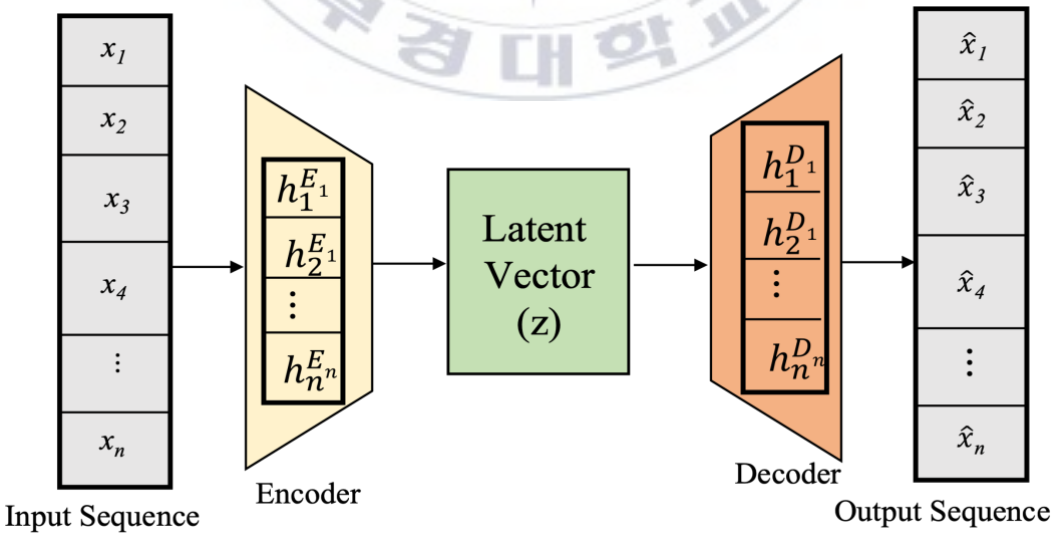


Figure 4.3: Schematic illustration of the autoencoder architecture.

Figure 4.3 illustrate the ability of an autoencoder to perform unsupervised encoding and decoding of input data with high efficiency, without the need for external guidance or labeling. The encoding process generates a compressed latent representation of the inputs via feature space reduction, and subsequently, the decoder utilizes this representation to reconstruct the initial inputs [52]. With this method, the autoencoder learns the most critical characteristics and patterns of the data without explicitly labeling it, providing a powerful tool for anomaly detection and image generation. The autoencoder consists of several components, such as an input block, encoder, latent space, decoder, and output block. The primary goal of the encoder is to transform high-dimensional input data x the form of a vector $[x \epsilon R^m]$ into a lower-dimensional feature space representation z by eliminating any irrelevant data, commonly referred to as noise, from the features space. The input equation is represented by Equation (8) as follows:

$$z = f(w_x + b) \quad (8)$$

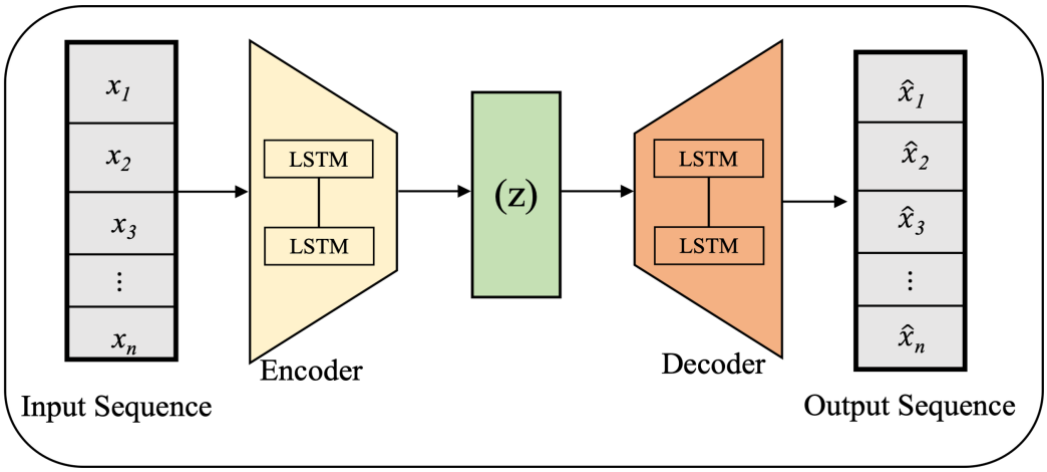
The activation function f operates on a low-dimensional data sequence z, with weight and bias denoted by w and b, respectively. The decoder takes the latent representation as input and transforms it to produce the reconstructed value of the original input data, denoted as $\hat{x}$. This transformation is represented by an output equation, which is defined as follows:

$$\hat{x} = f'(w'z + b') \quad (9)$$

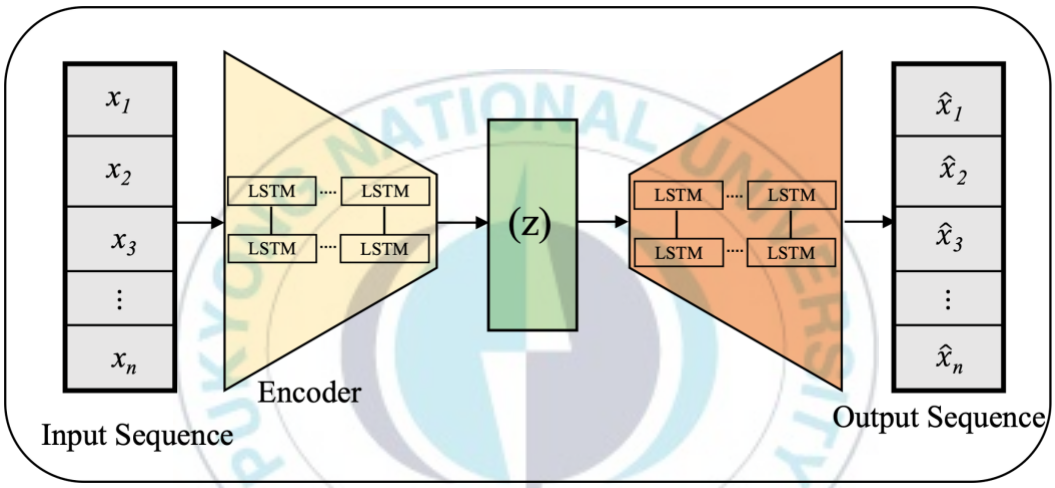
In the above equation, f' is the activation function, $w'$and b' represent the weight and bias of the reconstructed input $\hat{x}$ sample instance. In the standard autoencoder architecture, the objective is to minimize the reconstruction loss, which measures the dissimilarity between the input and output data. The loss function used for this purpose is the mean absolute error (MAE), which calculates the average absolute difference between the reconstructed output and the original input. The reconstruction loss can be expressed as follows:

$$L(x - \hat{x}) = \frac{1}{n}\sum_{n=1}^{n} /\hat{x}_t - x_t/ \quad (10)$$

The variable n represents the number of samples in the training dataset, while x and $\hat{x}$ denote the input and output data, respectively. To maximize the benefits of an autoencoder and an LSTM, the proposed model combines their capabilities. Important features are extracted from data using an autoencoder while the dimensionality of the features is reduced. These characteristics are then fed into an LSTM network, which identifies temporal dependencies. This method enables the autoencoder to function as a feature extractor, which can be used in conjunction with a supervised learning model to improve performance [53]. Figure 4.4 depicts the various model architectures used during the training process.



(a) 1- Layer LSTM



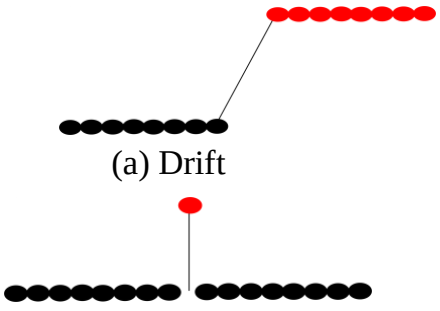
(b) N - Layer

Figure 4.4: Model training architectures

4.3 Anomaly Detection Approach

Following model training on a non-anomalous dataset, the anomaly classifier is utilized to assess the reconstructed data sample for abnormal events and identify them. The reconstruction loss is calculated using the mean absolute error (MAE) method, as shown in Equation (10). Alternative metrics utilized in the computation of reconstruction loss encompass the mean square error; nonetheless, the choice of MAE is strongly suggested because it has demonstrated

enhanced resilience in various studies and a reduced incidence of erroneous positive predictions generated by the model [54]. In order to establish the threshold for detecting anomalies, the highest value of MAE is utilized as the benchmark. Consequently, during instances where the sensor data conform to the standard behavior, the reconstruction loss calculated for the testing set falls below the determined anomaly threshold. When anomalies are detected within the data, the reconstruction loss exceeds the designated threshold. Typically, a sensor anomaly includes various types such as drift, gap, outlier, and noise. The drift anomaly type manifests as a continuous deviation in a single feature value within a time step or multiple time steps. Gaps and outliers occur when a value abruptly jumps across several time steps. Noise refers to deterministic and irregular variations in the dataset. In the context of this study, we specifically focus on drift and outlier anomalies, as depicted in Figure 4.5. These types of anomalies are frequently encountered in EPS systems, A comprehensive description of the approach pertaining to these anomalies is provided in chapter five.



(a) Drift

(b) Outlier

Figure 4.5: Patterns of detected anomalies in EPS torque sensor

CHAPTER FIVE

The chapter provides all the necessary information regarding experimental design, such as the data collection process, the training environment setup, model hyperparameters, detailed description of performance metrics. The result analysis and discussion.

5.1 Dataset Collection

To validate the effectiveness of the proposed approach, the torque sensor data of an EPS test jig is obtained. This objective is achieved with the aid of a 12-bit analog-to-digital converter to ensure precise and accurate measurements. Figure 5.1 depicts the schematic representation of the experimental setup. The operational speed ranged from 0 to 50 km/h, and the data acquisition board was implemented using a field-programmable gate array. A total of 87,555 data instances were collected by transferring the sample data at 10-ms intervals via the board's general-purpose input and output pins.

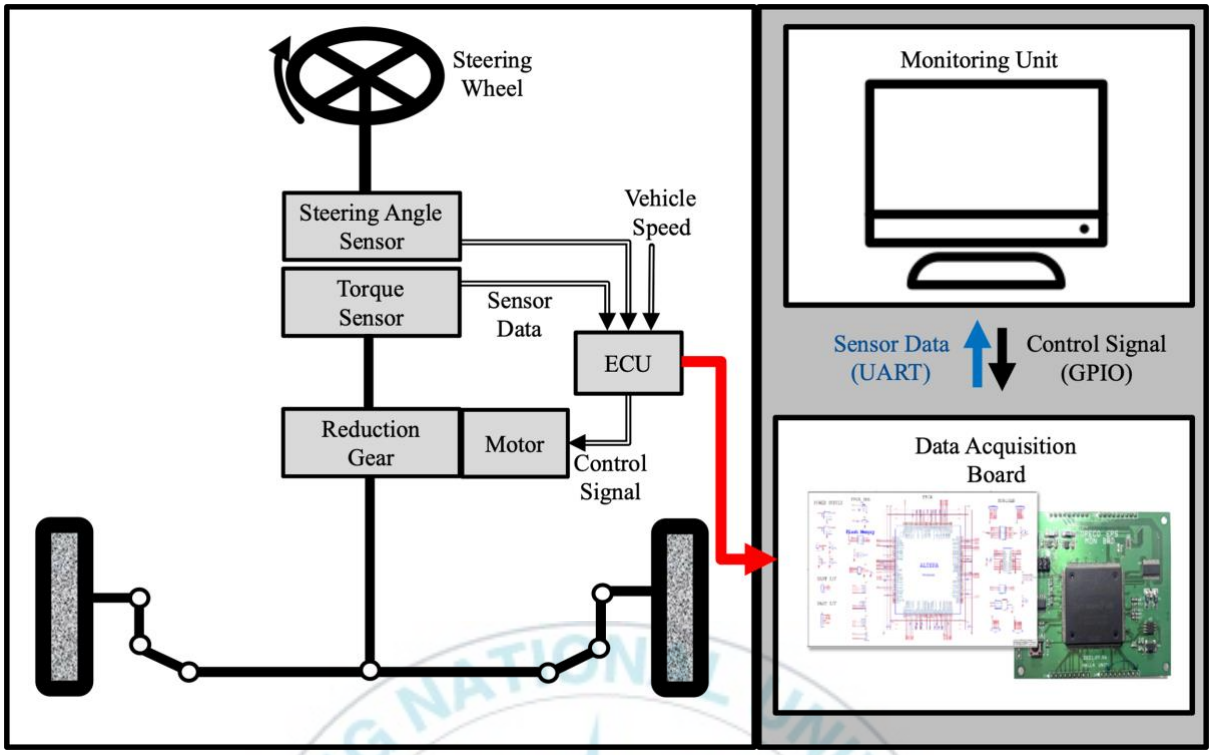


Figure 5.1: Schematic illustration of dataset collection

5.2 Experimental Setup and Model Hyperparameters

In this thesis, the model is trained on non-anomalous EPS torque sensor data sets. Obtaining data with anomalies is often challenging due to several factors. Firstly, anomalies are often rare events captured in a working system, making it difficult to have a sufficient number of instances for comprehensive training of models. Secondly, the annotation of dataset is expensive because it requires domain expert and time-consuming. A semi-supervised learning approach provides a solution to this setback through the use of normal dataset. This process of training with normal data has previously demonstrated adequate results with deep learning models [41,44,55]. However, the testing dataset

incorporated both normal and abnormal data. The dataset was partitioned into training, validation, and testing subsets before commencing the training of the model. To reduce the reconstruction loss, the training process employs both the training and validation datasets. This method can optimize the parameters based on the training dataset. Their generalizability and performance can be verified on the validation dataset at the same time. Deep learning models require this approach, as overfitting can undermine the generalization of new data when models are overtrained on the training dataset. As illustrated in Figure 5.2, the training and validation errors were plotted against the number of epochs to evaluate the effectiveness of model training on EPS sensor data. The graph shows rapid stabilization of the training loss at around 10 epochs, as well as subsequent stability of the validation loss. The hyperparameters employed for training the model are presented in Table 1, which includes an encoder and decoder, each with three LSTM layers and 128 units.

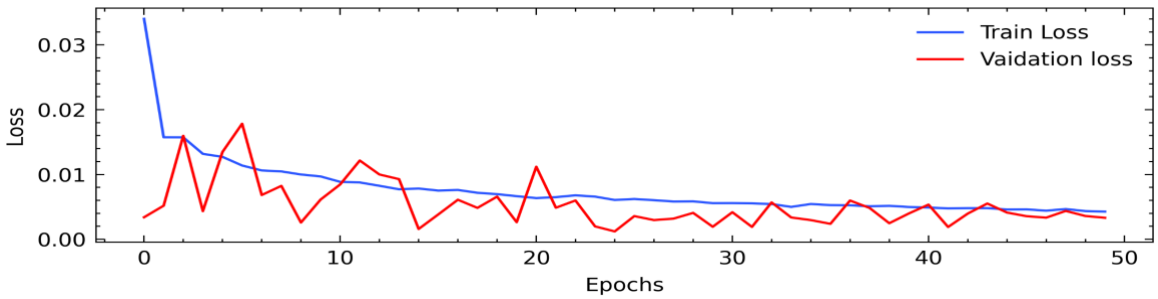


Figure 5.2: Epoch graph of training and validation loss for EPS torque sensor

Table 1. Model training Parameters

Parameter	**Value Description**
Model Framework	Pytorch
Layers	2
Learning Rate	0.0009
Optimizer	Adam [56]
Loss Function	MAE
Epoch size	50

5.3 Evaluation Metrics

Generally, in the field of deep learning, the utilization of evaluation metrics is a prevalent approach to ascertain whether the learning algorithm has met the desired or anticipated level of performance. In cases where the algorithm falls short of expectations, tuning measures may be necessary to optimize the model's objective. The analysis includes information on the models' ability to accurately identify real anomalies and differentiate them from false positives, as measured by sensitivity and specificity.

Thus, the comparative analysis of various algorithms based on benchmark evaluation provides valuable insights into selecting the most appropriate model.

The procedure for distinguishing between correct and incorrect classifications in relation to actual anomalies in a dataset is depicted in Table 2. This specific kind of table is referred to as a confusion matrix. True positive (TP) denotes the count of accurately detected anomalies, whereas true negative (TN) signifies the count of precisely detected normal data that does not contain any anomalies. The misclassification of anomalies is indicated by false positive (FP) refers to the percentage of normal data that is erroneously classified as anomalous, while false negative (FN) corresponds to the portion of anomalous data that is not classified as anomalous.

Table 2 provides an overview of the confusion matrix, when examining the output of an anomaly detection classification model.

		Anomaly Detection Model	
		1	0
Real Anomaly?	1	True positive	False negative
	0	False positive	True Negative

An evaluation of the proposed techniques is conducted using several metrics, including classification accuracy (A), precision (P), recall (R), and F-score (F). The accuracy of the anomaly detection algorithm is the proportion of correctly detected anomalies to all data in the dataset as shown in the following equation.

$$A = \frac{TP+TN}{TP+TN+FN+FP} \qquad (11)$$

Precision, as denoted by the equation below, is defined as the proportion of actual anomalies to predicted anomalies identified by the model. It characterizes the accuracy of identified anomalies and distinguishes them from false anomalies.

$$P = \frac{TP}{TP+FP} \tag{12}$$

Recall, represented by equation 13, denotes the proportion of anomalies forecasted by the model out of the entire set of anomalies. Sensitivity is a synonym used to describe recall and conveys the model's ability to detect the actual anomalies in the dataset.

$$R = \frac{TP}{TP+FN} \tag{13}$$

A comprehensive metric that balances the trade-off between precision and recall is the F-score, which is calculated as the harmonic mean using Equation 14. Through evaluation of the model's effectiveness in anomaly detection by taking into account both the accuracy of identified anomalies (precision) and the capacity to identify actual anomalies (recall).

$$R = 2 \cdot \frac{P \cdot R}{P+R} \tag{13}$$

Hence, higher accuracy signifies a more precise detection of normal and abnormal data by the model. Moreover, increased recall reflects a larger number of detected anomalies, while higher precision results in fewer false alarms. The experiment was conducted on the EPS dataset with three models, and the

corresponding metric scores are reported in Table 3. The model with the greatest performance analysis score is highlighted in bold letters.

Table 3. performance comparison considering the values of true positive (TP), false positive (FP), false negative (FN), and true negative (TN), as well as detection accuracy, precision, recall, and F1-score.

Model	TP	FP	FN	TN	Accuracy	Precision	Recall	F1-score
BiLSTM-AE	423	0	78	12,632	0.9950	0.9999	0.8443	0.9155
GRU-AE	446	0	55	12,632	0.9968	0.9999	0.8902	0.9419
LSTM-AE	**492**	**0**	**9**	**12,632**	**0.9993**	**0.9999**	**0.9823**	**0.9809**

5.4 Performance Results

5.4.1 EPS Data Anomaly Detection

The test datasets are employed to detect deviations or abnormalities by making inferences from the trained LSTM-AE model. The model's performance is gauged by calculating the reconstruction error. This measures the discrepancy between the model's reconstructed and original input data. A threshold for identifying anomalies was determined through Equation (10) in conjunction with the scaling magnitude approach described in [57]. The samples obtained from this process are labeled as ground truth anomalies and used to assess the model's ability to detect anomalies. Figure 5.3 visually presents the maximum acceptable range of the EPS torque sensor data, where the red dashed line represents the

benchmark, and the red spots represent the detected anomalies. Typically, the range of data scores is between 0.00 and 0.01.

The anomalies detected in the sensors encompassed various types as stated in chapter four, such as gap, drift, break, and outlier anomalies, with varying degrees of occurrence. This was achieved by scaling the magnitude value within a range of 1 to 1.5, where a magnitude of 1 denoted the absence of anomalies and a magnitude of 1.5 signified a 50% increase in the value. Drift anomalies and outliers were deemed to be present when alterations in the overall distribution or individual data points of the sample data were observed, as such changes could be indicative of mechanical degradation or deterioration, as described in reference [58].

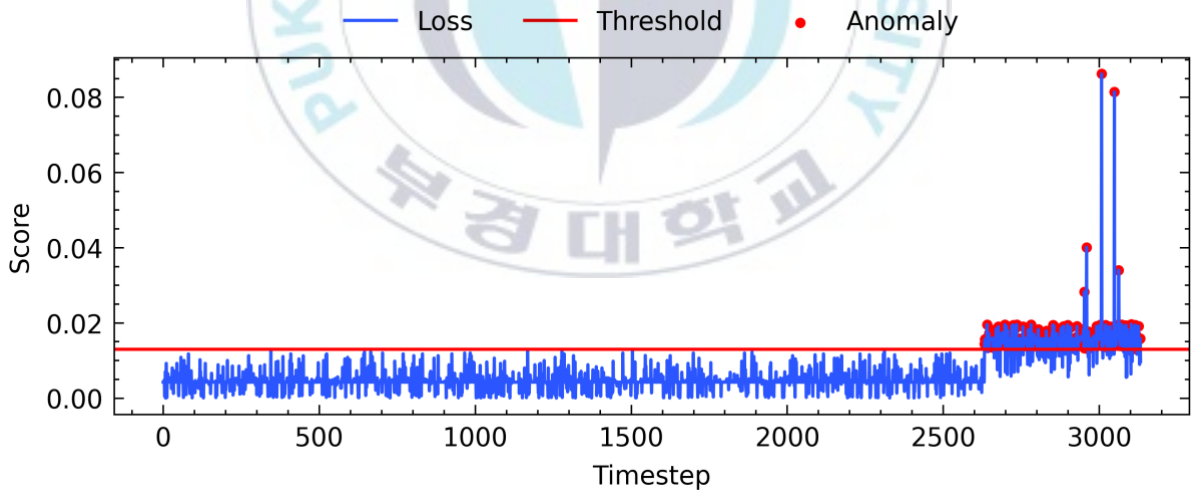
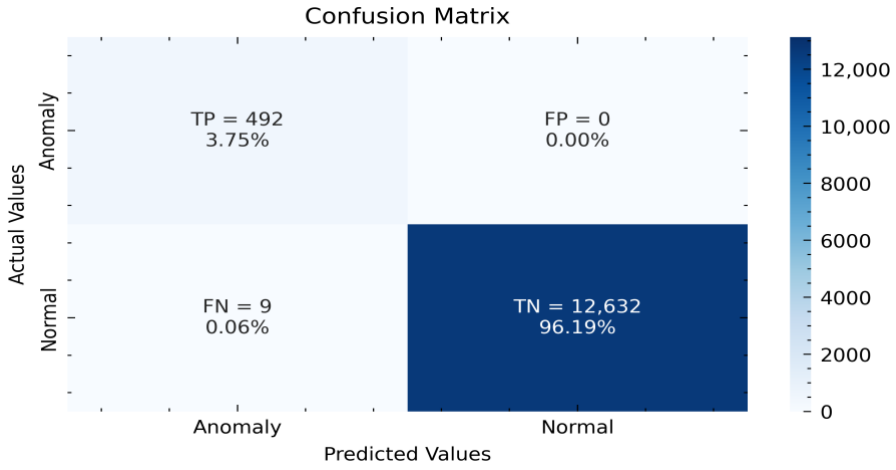


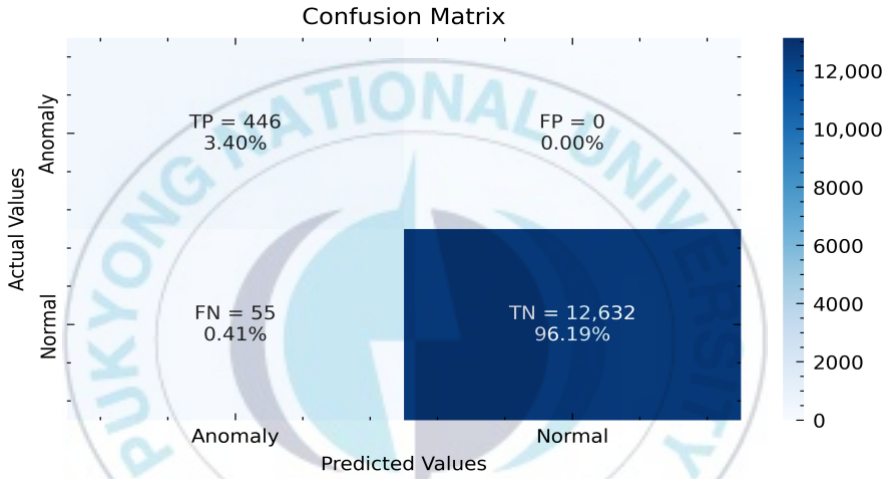
Figure 5.3: Anomaly detection on EPS torque sensor data

5.4.2 Models Result Analysis

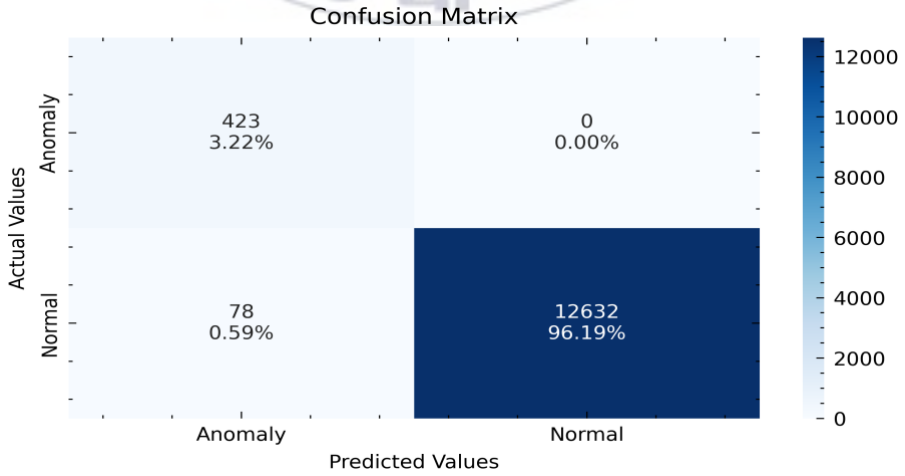
In order to assess the performance of the model, an analysis is conducted using the earlier discussed confusion matrix, which included the proposed approach, the gated recurrent unit autoencoder (GRU-AE), and the bi-directional long short-term memory autoencoder (BiLSTM-AE). As shown in Figure 5.3, most of the non-anomalous sample generalized between 0.01. Hence, the anomalous scenario is distinguished to be any instance above the MAE value. The confusion matrix for the test samples is illustrated in Figure 5.4. The total number of test samples amounts to 13,133, with 12,632 classified as non-anomalous samples and 501 classifieds as anomalous. The proposed approach (LSTM-AE) correctly recognized 492 aberrant data points out of a total of 501 abnormal samples, detecting 12,632 normal samples with an accuracy of 0.99. The model exhibited zero false positives, implying precise categorization of normal samples. However, there were nine false negatives, indicating the incorrect classification of abnormal samples as normal. GRU-AE detected 12,632 normal samples (specificity: 0.99) but identified 446 abnormal samples (sensitivity: 0.89) out of 501 samples. Additionally, the Bi-LSTM model effectively identified a total of 12,632 normal samples with a specificity of 0.99 and only 423 aberrant data points with a sensitivity of 0.84 out of a total of 501 anomalous samples. Table 3 provides a summary of the models' performance.



(a). LSTM-AE Model



(b). GRU-AE Model



(c). BiLSTM-AE Model

Figure 5.4: Confusion matrix of model detection on EPS data

Furthermore, a graphical representation of the Area Under the Curve (AUC) is generated to assess the discriminative performance of the three models on EPS toque sensor, as illustrated in Figure 5.5. AUC serves as a quantitative measure of the distinction between the True Positive (TP) and False Positive (FP) rates in the LSTM-AE model. The plotted curve explicitly exhibits the high anomaly detection capability of our model, with a remarkable AUC score of 0.99. The GRU-AE model exhibited a slightly inferior performance with a score of 0.95 compared to the LSTM-AE model, primarily attributed to the absence of regulatory memory cells. In contrast, the LSTM unit's output gate effectively governs the extent of memory content that is accessible or utilized by other network units, contributing to the superior performance of the LSTM-AE model. GRU, due to its fewer training parameters, is characterized by lower memory usage and faster execution compared to LSTM. Conversely, LSTM tends to demonstrate higher accuracy when dealing with larger datasets, owing to its inherent capability to capture long-term dependencies.

The BiLSTM-AE model yielded a score of 0.92, albeit with a relatively lower True Positive (TP) rate. The contrasting performance between BiLSTM-AE and LSTM-AE models can be attributed to the direction of information flow. Unlike LSTM-AE, which processes input sequentially from past to future, BiLSTM-AE takes into account both directions by processing input in both

forward and backward directions during network training. This bidirectional processing approach of BiLSTM-AE could potentially impact the TP rate and overall performance of the model. Training a BiLSTM incurs higher computational costs compared to a unidirectional LSTM. Furthermore, a larger amount of data is typically necessary to achieve performance levels comparable to those of a unidirectional LSTM. This result analysis shows the robustness of the threshold selection and quantitative evaluation of the model's performance to enable fair comparison on the EPS dataset for anomaly detection. To evaluate the efficiency of the model in comparison to similar anomaly detection techniques that utilize distinctive features of LSTM, AE, or a fusion of both, Table 4 provides a comprehensive assessment of their respective performances using metrics such as model accuracy, precision, recall, and F1-score.

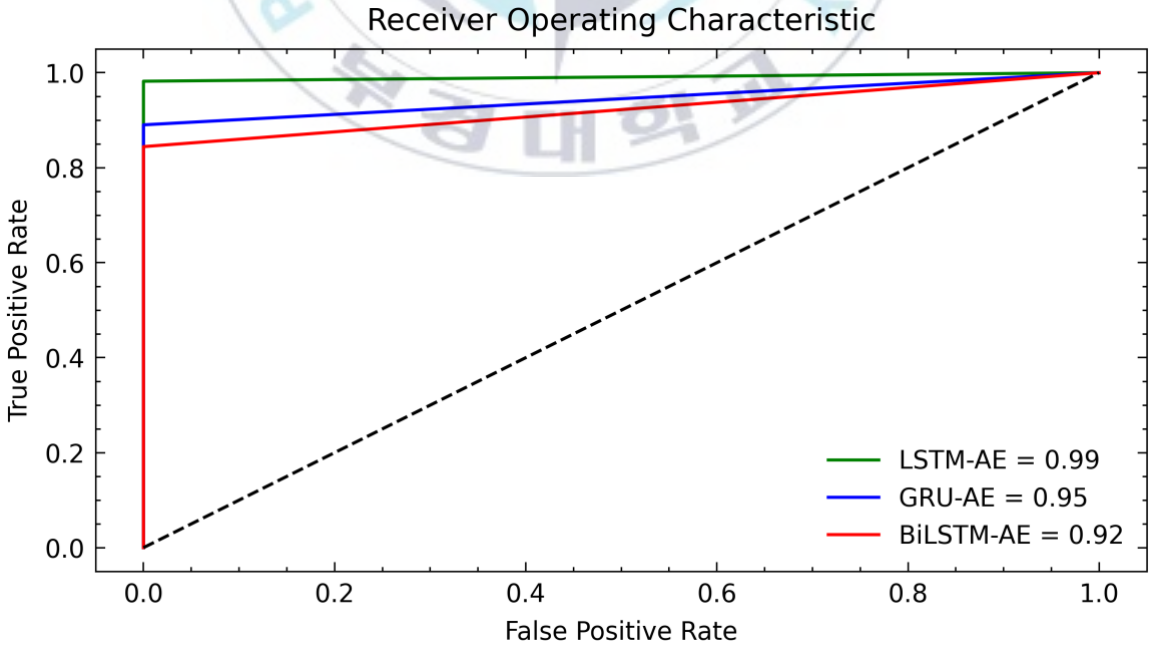


Figure 5.5: Model receiver operation characteristic curve

Table 4: Benchmarking performance against similar models

Model	Dataset	Source	Accuracy	Precision	Recall	F1-score
LSTM-AE	IPC-SHM2020	[44]	0.9998	0.9568	0.9201	0.9382
C-LSTM	Web S5	[39]	98.6	96.2	89.7	92.3
LSTM-AE	ECG	[53]	98.57	97.74	98.85	Nil
LSTM-AE	Solar plant	[59]	0.8963	0.9474	0.9432	0.9453
BiLSTM-AE	Smart meter	[60]	0.9957	0.9958	0.9999	0.9978
LSTM-AE	BOU	[54]	0.9444	0.9794	0.5877	0.9145
BiLSTM-AE	UNM	[61]	90.01	84259	97.87	90.75
LSTM-AE	EPS	Ours	0.9993	0.9999	0.9820	0.9809

By comparing the values of these metrics across different models that applied the distinct feature of LSTM or AE, we can assess the relative performance and effectiveness of the models in anomaly detection for sensor datasets. For example, the LSTM-AE model achieves high metrics scores in the ECG dataset, solar plant, structural health monitoring, and brake operation unit indicating its effectiveness in detecting anomalies. Similarly, the BiLSTM-AE model performs exceptionally well in the Smart meter dataset and UNM dataset. These benchmark comparisons help ascertain the efficiency of the proposed approach for EPS torque sensor anomaly detection.

CHAPTER SIX

This chapter concludes the thesis and describes the direction for future works.

6.1 Conclusion and Future Work

This study proposes an innovative anomaly detection framework for EPS sensor devices utilizing an LSTM-based autoencoder. The autoencoder includes an encoding algorithm that captures the unique input data representation while being implemented with an LSTM network to accurately simulate the dataset's temporal dependencies. This strategy eliminates the requirement for significant feature engineering expertise or intricate preprocessing techniques to extract meaningful features from the sensor signal. The proposed framework adopts a semi-supervised learning approach, which solely necessitates training the model with normal data, thus alleviating the burden of obtaining labeled data for anomaly detection.

Hence, the proposed framework effectively captures the anticipated distribution of the sensor signal, with the maximum mean absolute error of the reconstruction loss from the trained model serving as the anomaly score for inference on test data samples. The performance of the proposed model is evidenced by its accurate detection of simulated drift and outlier anomalies, validating its robustness in anomaly detection tasks. This study simulates drift and outlier anomaly events using EPS torque sensor data. Future research needs to explore real-world anomalous conditions collected from EPS systems,

including anomalies such as gaps in non-contact torque sensor coil signals with significant differences between primary and auxiliary signals and a sharp drop from higher to lower magnitude in speed sensor readings. These investigations aim to enhance the resilience of the proposed framework in identifying anomalies in EPS systems.

Reference

[1] M. Würges, “New Electrical Power Steering Systems,” Encyclopedia of Automotive Engineering, pp. 1–17, 2013.

[2] LEARNING MODEL: Electric Power Steering. In Proceedings of the Exxotest Education, 2007, pp.1–45, 2022.

[3] GreyViews, “Electric Power Steering Market- Segmentation and Projection (2022-2029),” GreyViews. https://greyviews.com/reports/electric-power-steering-market/54 , 2021

[4] K.-J. Lee, K.-H. Lee, C. Moon, H.-J. Chang, and H.-S. Ahn, “Design and Development of a Functional Safety Compliant Electric Power Steering System,” Journal of Electrical Engineering and Technology, vol. 10, no. 4, pp. 1915–1920, Jul. 2015.

[5] X. Wang, Y. Zhao, and H. Wang, "Non-conduct steering sensor for Electric Power Steering," in Proceedings of the 2009 International Conference on Information and Automation, pp. 1-4, 2009.

[6] J. Lee, H. Lee, J. Kim, and J. Jeong. "Model-based fault detection and isolation for electric power steering system." In 2007 International Conference on Control, Automation and Systems, pp. 2369-2374, 2007.

[7] W. -C. Lin and Y. A. Ghoneim, "Model-based fault diagnosis and prognosis for Electric Power Steering systems," 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 1-8, 2016.

[8] A. A. Cook, G. Mısırlı and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6481-6494, 2020.

[9] W. Yan and L. Yu. "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach, pp.1-8, 2019.

[10] A. D. Jasim. "A survey of intrusion detection using deep learning in the Internet of Things." Iraqi Journal for Computer Science and Mathematics 3.1, 83-93, 2022.

[11] S. Namuduri, B. N. Narayanan, V. S. P. Davuluru, L. Burton, and S. Bhansali, “Review—Deep Learning Methods for Sensor Based Predictive Maintenance and Future Perspectives for Electrochemical Sensors,” Journal of The Electrochemical Society, vol. 167, no. 3, pp. 037552, 2020.

[12] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, pp. 1-6,2018.

[13] M. Batta, Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, pp.381-386, 2020.

[14] M. Mohammed, M. B. Khan, and E. B. M. Bashier. "Machine learning: algorithms and applications." CRC Press, 2016.

[15] S. Angra and S. Ahuja, "Machine learning and its applications: A review," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, Andhra Pradesh, India, pp. 57-60, 2017.

[16] I. H. Sarker, A. S. M. Kayes, S. Badsha, et al. "Cybersecurity data science: an overview from machine learning perspective." Journal of Big Data vol.7, pp.1-29, 2020.

[17] S. Russel and P. Norvig. "Artificial intelligence: a modern approach.", London: Pearson Education Limited, 2013.

[18] M. Usama, J. Qadir, A. Raza, H. Arif, K. L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha. "Unsupervised machine learning for networking: Techniques, applications and research challenges." IEEE Access 7, pp. 65579-65615, 2019.

[19] I. H. Sarker. "Machine learning: Algorithms, real-world applications, and research directions." SN Computer Science 2(3), p.160, 2021.

[20] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[21] N. K. Chauhan and K. Singh, "A Review on Conventional Machine Learning vs Deep Learning," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, pp. 347-352, 2018

[22] Y. Bengio, and A. Courville. "Deep learning." MIT Press, 2016.

[23] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey." ACM Computing Surveys (CSUR) vol. 41, no. 3, 2009.

[24] M. Ahmed, A. N. Mahmood, and J. Hu. "A survey of network anomaly detection techniques." Journal of Network and Computer Applications, pp. 19-31, 2016.

[25] K. Ando, A. Murohashi, M. Fujikake, and T. Horikawa. "Development of electric power steering evaluation system." KYB Technical Review, Vol. 56, pp. 37-42, 2018.

[26] T. Salman, D. Bhamare, A. Erbad, R. Jain and M. Samaka, "Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments," 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, pp. 97-103, 2017.

[27] D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta and C. C. Kemp, "A multimodal execution monitor with anomaly classification for robot-assisted feeding," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, pp. 5406-5413,2017.

[28] N. V. Chawla, N. Japkowicz, and A. Kotcz. "Special issue on learning from imbalanced data sets." ACM SIGKDD Explorations Newsletter 6, pp. 1-6, 2004.

[29] H. S. Emadi and S. M. Mazinani. "A novel anomaly detection algorithm using DBSCAN and SVM in wireless sensor networks." Wireless Personal Communications, pp. 2025-2035, 2018.

[30] S. Mishra and M. Chawla. "A comparative study of local outlier factor algorithms for outliers detection in data streams." In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Vol 2, pp. 347-356, 2019.

[31] Z. Liu, T. Qin, X. Guan, H. Jiang and C. Wang, "An Integrated Method for Anomaly Detection From Massive System Logs," in IEEE Access, vol. 6, pp. 30602-30611, 2018.

[32] G. Münz, S. Li, and G. Carle. "Traffic anomaly detection using k-means clustering." In GI/ITG Workshop MMBnet, vol. 7, no. 9, 2007.

[33] J. MacQueen. "Classification and analysis of multivariate observations." In 5th Berkeley Symp. Math. Statist. Probability, pp. 281-297. University of California, Los Angeles, LA, USA, 1967.

[34] I. T. Jolliffe and J. Cadima. "Principal component analysis: a review and recent developments." Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, pp. 1-16, 2016.

[35] T. Kudo, T. Morita, T. Matsuda and T. Takine, "PCA-based robust anomaly detection using periodic traffic behavior," 2013 IEEE International Conference on Communications Workshops (ICC), Budapest, Hungary, pp. 1330-1334, 2013

[36] L. Ben Amor, I. Lahyani and M. Jmaiel, "PCA-based multivariate anomaly detection in mobile healthcare applications," 2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Rome, Italy, pp. 1-8, 2017.

[37] Y. Wang, C. Yang and W. Shen, "A Deep Learning Approach for Heating and Cooling Equipment Monitoring," 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, pp. 228-234, 2019.

[38] R. Zhao, J. Wang, R. Yan and K. Mao, "Machine health monitoring with LSTM networks," 2016 10th International Conference on Sensing Technology (ICST), Nanjing, China, pp. 1-6, 2016.

[39] T.-Y. Kim and S.-B. Cho. "Web traffic anomaly detection using C-LSTM neural networks." Expert Systems with Applications, pp. 66-76, 2018.

[40] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection." In International Conference on Learning Representations, pp. 1-19, 2018.

[41] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection." arXiv preprint arXiv:1607.00148, 2016.

[42] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras and P. Sarigiannidis, "A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1137-1151, 2021.

[43] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, 2018.

[44] H. Son, Y. Jang, S. -E. Kim, D. Kim and J. -W. Park, "Deep Learning-Based Anomaly Detection to Classify Inaccurate Data and Damaged Condition of a Cable-Stayed Bridge," in IEEE Access, vol. 9, pp. 124549-124559,

[45] H. Jo, S.-H. Sim, K. A. Mechitov, R. Kim, J. Li, P. Moinzadeh, B. F. Spencer Jr, et al. "Hybrid wireless smart sensor network for full-scale structural health monitoring of a cable-stayed bridge." In Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2011, vol. 7981, pp. 45-59, 2011.

[46] S. Na, Z. Li, F. Qiu, and C. Zhang. "Torque control of electric power steering systems based on improved active disturbance rejection control." Mathematical Problems in Engineering, pg. 1-13, 2020

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12, pg. 2825-2830, 2011

[48] P. J. M. Ali, R. H. Faraj, E. Koya, P. J. M. Ali, and R. H. Faraj. "Data normalization and standardization: a technical report." Machine Learning Technical Report 1, no. 1 pg. 1-6, 2014.

[49] S. Hochreiter and J. Schmidhuber. "Long short-term memory." Neural Computation 9(8), pp. 1735-1780, 1997.

[50] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, 2017.

[51] F. Landi, L. Baraldi, M. Cornia, and R. Cucchiara. "Working memory connections for LSTM." Neural Networks 144, pp. 334-341, 2021.

[52] B. Siegel, "Industrial Anomaly Detection: A Comparison of Unsupervised Neural Network Architectures," in IEEE Sensors Letters, vol. 4, no. 8, pp. 1-4, 2020.

[53] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, and C. Wu. "Arrhythmia classification of LSTM autoencoder based on time series anomaly detection." Biomedical Signal Processing and Control 71, pp. 103228, 2022.

[54] J. Kang, C.S. Kim, J.W. Kang, and J. Gwak. "Anomaly detection of the brake operating unit on metro vehicles using a one-class LSTM autoencoder." Applied Sciences 11(19), pp. 9290, 2021.

[55] P. Han, A. L. Ellefsen, G. Li, F. T. Holmeset and H. Zhang, "Fault Detection With LSTM-Based Variational Autoencoder for Maritime Components," in IEEE Sensors Journal, vol. 21, no. 19, pp. 21903-21912, 2021.

[56] D.P. Kingma and J. Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.

[57] A. Liebert, W. Weber, S. Reif, B. Zimmering and O. Niggemann, "Anomaly Detection with Autoencoders as a Tool for Detecting Sensor Malfunctions," 2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS), pp. 01-08, 2022.

[58] L.W. Zhang, R. Du, and Y. Cheng. "The analysis of the fault of electrical power steering." In MATEC Web of Conferences, vol. 44, p. 02003. EDP Sciences, 2016.

[59] M. Ibrahim, A. Alsheikh, F.M. Awaysheh, and M.D. Alshehri. "Machine learning schemes for anomaly detection in solar power plants." Energies 15, no. 3, pp. 1082, 2022.

[60] S. Lee, H. Jin, S.H. Nengroo, Y. Doh, C. Lee, T. Heo, and D. Har. "Smart Metering System Capable of Anomaly Detection by Bi-directional LSTM Autoencoder." In 2022 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6, 2022.

[61] Y. Wang, X. Chen, Q. Wang, R. Yang and B. Xin, "Unsupervised Anomaly Detection for Container Cloud Via BILSTM-Based Variational Auto-Encoder," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3024-3028, 2022.

Publications

(1). Journal

Paper Title	Date	Journal Title
A Deep Learning Approach to Detect Anomalies an Electric Power Steering System	Nov. 2022	Sensors
Design and Implementation of a Data-Driven Defect and Linearity Assessment Monitoring System for Electric Power Steering	April 2023	Journal of Internet of Things and Convergence

(2). Conference

Paper Title	Date	Conference Title
Design and Implement of EPS Defect Monitoring System	June 2022	대한전자공학회 하계학술대회