SECURITY ANALYSIS AND PERFORMANCE EVALUATION OF A COMBINED CNN-LSTM WITH SELF-SIMILARITY AND HURST PARAMETER FOR ICS TRAFFIC

BY

ASAAD BALLA FADL ELMULA BABIKER

A thesis submitted in fulfilment of the requirements for the degree of Master of Science in Engineering (MSE)

Kulliyyah of Engineering
International Islamic University Malaysia

JANUARY 2024

ABSTRACT

As the integration of IoT devices with SCADA systems increases, concerns about cyber security have become significant. This thesis addresses the challenge of data imbalance in developing an effective intrusion detection system (IDS) for SCADA systems. To tackle this issue, we employ the DeepInsight package in Python to convert traffic data into grayscale images. Four publicly available SCADA datasets are analyzed using exploratory data analysis (EDA) and principal component analysis (PCA). Our research evaluates two detectors: the first utilizes the Hurst parameter to differentiate between normal and attack image data, while the second employs a state-of-the-art CNN-LSTM algorithm—the Hurst Detector leverages self-similarity to identify abnormal network traffic data in conjunction with the CNN-LSTM model. For feature extraction, we propose a CNN and PCA approach applied to the converted grayscale images of the Morris Power dataset. The model includes input, hidden, and output layers with activation functions, while the RNN LSTM modifies the LSTM, dense, and output layers by incorporating appropriate activation functions. Additional layers for Batch Normalization (BN) and dropout enhance the model's performance. The performance of the detectors is evaluated using standard metrics, including accuracy, precision, recall, and F1-score. Results indicate that the combination of self-similarity Hurst index and Deep Learning (DL) achieves a detection accuracy of 98.2% for attacks, while the combined detectors utilizing CNN-LSTM achieve an accuracy of 99.92%. These findings provide valuable insights for security researchers and practitioners seeking to enhance cyber security in SCADA systems. Through an enhanced approach, this DL model has the potential to strengthen SCADA system security and effectively mitigate cyber attacks.

ملخص البحث

مع زيادة تكامل أجهزة الإنترنت من الأشياء مع أنظمة SCADA، أصبحت المخاوف المتعلقة بالأمان السيبراني مهمة بشكل كبير. تتناول هذا الأطروحة تحدي التوازن في البيانات أثناء تطوير نظام كشف الاختراق الفعّال (IDS) لأنظمة SCADA. لمواجهة هذه المشكلة، نستخدم حزمة DeepInsight في لغة البرمجة Python لتحويل بيانات حركة المرور إلى صور باللون الرمادي. يتم تحليل أربع مجموعات بيانات SCADA متاحة للعموم باستخدام تحليل البيانات الإستكشافي (EDA) وتحليل المكونات الأساسية (PCA). يقوم بحثنا بتقييم مُكتَشِفَيْن: الأول يستخدم معامل هيرست للتمييز بين البيانات العادية والهجوم، بينما يستخدم الثاني خوارزمية CNN-LSTM عصرية. يستفيد مكتشف هيرست من التشابه الذاتي لتحديد بيانات حركة المرور غير العادية بالتعاون مع نموذج CNN-LSTM. بالنسبة لاستخراج الميزات، نقترح نهجًا يعتمد على CNN و PCA يُطبَّق على صور اللون الرمادي المحولة من مجموعة بيانات Morris Power. يتضمن النموذج طبقات الإدخال والطبقات الخفية والإخراج بوظائف التنشيط، بينما يُعد نموذج الشبكات العصبية العميقة LSTM تعديلًا لطبقات LSTM والكثافة والإخراج باستخدام وظائف التنشيط ذات الصلة. تعزز الطبقات الإضافية للتوحيد التسلسلي (BN) وإسقاط القيمة الزائدة من أداء النموذج. يتم تقييم أداء المكتشفين باستخدام معايير قياسية، بما في ذلك الدقة والصحة والاستدعاء ونسبة الف1. تشير النتائج إلى أن توحيد مؤشر التشابه الذاتي للهيرست والتعلم العميق يحقق دقة كشف بنسبة 98.2% للهجمات، بينما يحقق المكتشفان المدمَجان باستخدام CNN-LSTM دقة بنسبة 99.92%. تقدم هذه النتائج رؤى قيمة للباحثين والممارسين في مجال تعزيز الأمان السيبراني في أنظمة SCADA. من خلال نهج محسَّن، يتمتع هذا النموذج التعلم العميق بالقدرة على تعزيز أمان نظام SCADA بشكل كبير.

APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Engineering

Mohamed Hadi Habaebi
Supervisor

Mohammad Rafiqul Islam
Co-Supervisor

Farah Diyana Bt. Abdul Rahman
Co-Supervisor

I certify that I have read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Engineering.

Name
Internal Examiner

Name
External Examiner

This thesis was submitted to the Department of Electrical and Computer Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science in Engineering

Mohammad Rafiqul Islam
Head, Department of Electrical and
Computer Engineering

This thesis was submitted to the Kulliyyah of Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science in Engineering.

Sany Izan Ihsan
Dean, Kulliyyah of Engineering

DECLARATION

I hereby declare that this dissertation is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Asaad Babiker

Signature: Date: September 27, 2023

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

A DEEP LEARNING MODEL FOR CYBER-PHYSICAL SECURITY IN ICS SCADA SYSTEMS

I declare that the copyright holder of this thesis are jointly owned by the student and IIUM.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may only be used by others in their writing with due acknowledgement.

2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purpose.

3. The IIUM library will have the right to make, store in a retrieval system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Asaad Balla FadlElmula Babiker

…………………………	September 27 2023
Signature	Date

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

AI	Artificial intelligence
CI	Critical infrastructure
CIA	Confidentiality, integrity, availability
CNN	Convolutional neural network
DL	Deep learning
DoS	Denial of service
HMI	Human-machine interface
HTTP	Hypertext transfer protocol
ICS	Industrial control system
IDS	Intrusion detection system
IoT	Internet of things
IP	Internet protocol
IT	Information technology
ML	Machine learning
MTU	Master terminal unit
PLC	Programmable logic controller
RNN	Recurrent neural network Long-term memory
LSTM	
RTU	Remote terminal unit
SCADA	Supervisory control and data acquisition
SVM	Support vector machine
TCP	Transmission control protocol
UDP	User datagram protocol

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

A general term for various control system types, Industrial Control System (ICS) includes Supervisory Control and Data Acquisition (SCADA) systems, distributed control systems (DCS), and other control system configurations like Programmable Logic Controllers (PLC), which are frequently found in critical infrastructures and the industrial sector. Using specialized control systems like a Master Terminal Unit (MTU) and Remote Terminal Unit (RTU), SCADA systems are systems that a) monitor and control properties across a large geographic region and b) automate and control operations in the industrial sector (Riis, 2016). SCADA systems are used in various industries, including power generation, manufacturing, water treatment, and transportation. They are essential for many critical infrastructure systems' efficient and safe operation.

Internet of Things (IoT) devices are being integrated into current SCADA systems considering Industry 4.0, leading to Industrial IoT (IIoT). These technologies enable the operator to continuously monitor the machine's status and provide immediate feedback and changes. Additionally, integrating IoT into SCADA systems can enable the use of advanced analytics and machine learning algorithms to analyze data from the system and identify trends and patterns that can help optimize the industrial process. This can help organizations make more informed decisions and improve their overall operations. As a result, it can help industries that use a variety of electrical machinery, particularly induction motors, run more efficiently (Tran, 2021). Integrating IoT into

SCADA systems can bring many benefits. Still, it is essential to carefully consider the potential security risks and implement appropriate measures to protect the system.

SCADA systems are mainly composed of control centers and a variety of decentralized remote-field types of equipment, such as remote terminal units (RTUs), programmable logic controllers (PLCs), and Machine Interfaces (HMI), all of which are linked to a specific form of communication (Rezai, 2013). Figure 1.1 illustrates how these components work together in the SCADA architecture. Modernized SCADA systems are complex, complicated, and reliant on sophisticated technological systems.

SCADA systems are vulnerable to security threats. These threats can come from various sources, including hackers, malware, and insider threats. As a result, it is essential to implement strong security measures to protect SCADA systems from these threats. Using open standard protocols improves the productivity and profitability of SCADA systems. SCADA architecture significantly improves data access and is cost-effective, adaptable, adjustable, accessible, and scalable (Teixeira, 2018).

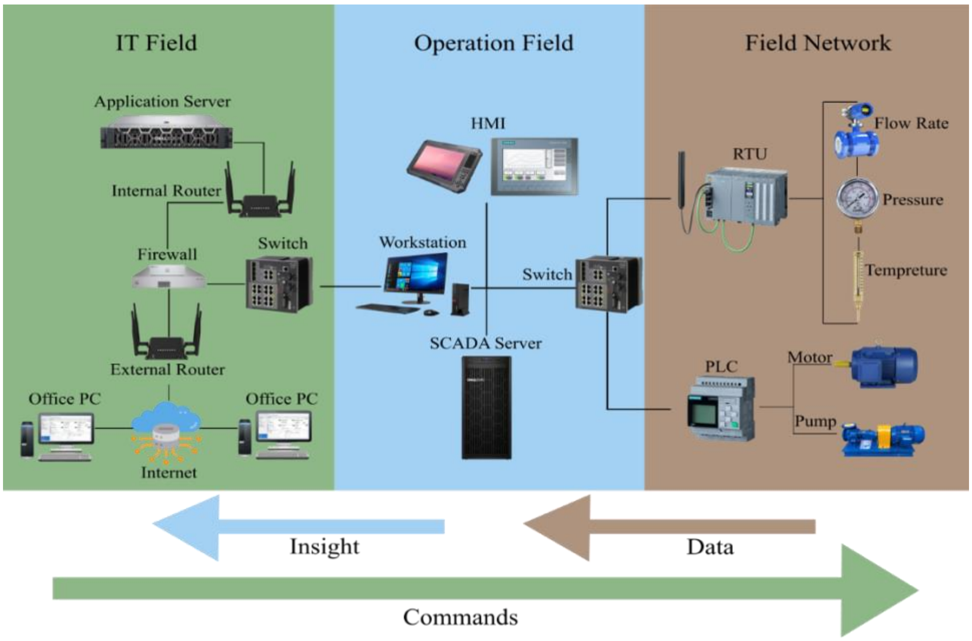


Figure 1.1 SCADA layered architecture.

SCADA system performance is vital for industries such as electricity, water, and transportation, where data collection, monitoring, and control are critical (Cherdantseva et al. 2016). SCADA systems perform monitoring, data logging, alarming, and diagnostic duties, as well as improve operational efficiency, reduce costs, and consume less energy.

Cyber security solutions for information technology sectors are well-developed and robust, but the work on cyber security for industrial control systems has been limited. The CIA triad, or confidentiality, integrity, and availability, is designed to drive information security practices within an organization. The top concern in the IT industry is confidentiality, while the absolute priority in the SCADA sector is availability. SCADA systems lack a cyber security culture, making them increasingly vulnerable to cyber attack vectors as they become more digital. Applications and standards in the SCADA domain are customized in SCADA, HMI, and DCS, whereas email, internet, video, and so on are already standardized in the IT domain. SCADA networks have inconsistencies in security policies and standards, and Industrial IoT IIoT/OT security still needs much work (Barracuda, 2022).

Historically, SCADAs have been used at remote sites. The traditional type of SCADA focuses on system functionality. Its design did not take data and network security into account. This concept, however, has proven too expensive to distribute, maintain, and run remotely. With the advancement of Information and Communications Technology (ICT) and functional needs, more SCADAs have transitioned from isolated systems to a public network for remote management and oversight of facilities (Asghar et al. 2019). The advancements of the SCADA systems resulted in new threats and vulnerabilities (Ten, 2008). This is highlighted again by other researchers

(Cherdantseva, 2016), indicating that the rapid evolution and automation, as well as real-time continuous operation and decentralized, multi-component design, are driving the increase of cyber attacks on SCADA systems. (Maglaras, 2018) noted that using the internet for communications in SCADA systems has contributed to the situation's complexity and severity.

SCADA systems have adopted several cyber-security strategies and devices from the information technology sector to address these challenges. Some standard security measures for SCADA systems include:

- Encrypting communications between the central computer, the sensors, and other devices in the system.
- Implementing strong authentication and access control measures to prevent unauthorized access to the system.
- Regularly applying security patches and updates to the system to fix vulnerabilities.
- Conducting regular security assessments and audits to identify and address potential vulnerabilities.
- Implement robust backup and disaster recovery plans if the system is compromised or fails.

The technique of observing and analyzing activities in Critical Infrastructure (CI) to detect evidence of security concerns is known as intrusion detection. To secure networks from cyber threats, the Intrusion Detection System (IDS) can be implemented with other security measures, such as access control, authentication protocols, and encryption methods. There are three broad categories: miss-used, anomaly-based, and hybrid. The misuse-based approach can easily detect intrusions that fit at least one dataset signature and have a low false-positive rate (Maglaras, 2018).

The rising prominence of intrusion detection methods has opened an exciting study topic in security and has enhanced the recognition of anomalies in SCADA systems. A more practical method of analyzing intrusion detection threats is using Deep Learning (DL) techniques, which have concentrated more on the exploits of typical IT vulnerabilities in SCADA. Artificial Intelligence (AI) offers various computing methods. One such method is ML techniques, which are suitable for Intrusion detection and can help humans detect and prevent increasing cyber-crimes (Perez et al. 2018). While some studies have been undertaken to date to review ML and Data Mining (DM) techniques for intrusion detection or malware detection, only a few of them included an overview of DL methods for intrusion detection. At the same time, there is no survey of DL techniques for malware detection or phishing detection (Ferrag et al., 2020).

1.2 PROBLEM STATEMENT

SCADA serves as the foundation for implementing cyber security solutions in ICS. The use of open standard protocols increased the productivity and profitability of the SCADA systems. SCADA architecture significantly enhances data access and is cost-effective, flexible, configurable, accessible, and scalable (Teixeira, 2018). However, these advancements created new threats and vulnerabilities (Ten, 2008). Cyber security solutions for information IT, like firewalls, intrusion detection, and intrusion protection systems, are ineffective for SCADA vulnerabilities. Current IDSs are trained using outdated SCADA datasets, which causes the issue of overfitting. This is due to the SCADA suppliers' concern about disclosing vulnerabilities in their infrastructure. Despite security precautions, there have lately been constant attacks on civilian and military SCADA infrastructures such as nuclear power stations, water treatment facilities, industrial facilities, and oil and gas operations, to mention a few. Furthermore, due to the sensitivity and criticality of industrial assets, the industry hesitates to share proprietary and sensitive functional data for public study.

The existing limitation of the lack of real-time datasets encouraged us to improve current algorithm efficiency and accuracy by investigating and proposing a DL model that can accommodate dataset imbalance. Dataset imbalance refers to a situation in which the dataset used to train the IDS contains a disproportionate number of examples of one class of data (e.g., normal behavior) compared to another (e.g., malicious behavior). Traditional ML algorithms may detect cyber attack anomalies. Still, the main challenge with present ML methods is that they cannot retrieve the essential features from network packets required to detect the complex nature of zero-day attacks. There has been little research on the advantages of integrating DL algorithms in SCADA systems. Security researchers should look at the potential of novel DL techniques for anomaly detection in SCADA systems.

DL models have been trending in recent years due to their ability to classify attacks accurately with low false alarms. A Deep Belief Network (DBN) and Probabilistic Neural Network (PNN) intrusion detection technique is proposed by (Zhao, Zhang, and Zheng 2017), (Wu, Chen, and Li 2018) presented a Convolutional Neural Network (CNN) method for malware classification by converting the network traffic data into images, and in (Su et al. 2020), a BLSTM-RNN model is proposed for intrusion detection.

This research focuses on improving anomaly detection in SCADA IDSs by addressing the influence of dataset imbalance. To address the issue, we used a DL CNN approach with image classification and a Recurrent neural network RNN LSTM (Long short-term memory) DL method for detecting abnormalities in SCADA datasets.

CNNs are a type of artificial neural network that is often used in the field of computer vision. They are particularly well-suited to tasks such as image classification and object detection. They have been applied to various applications in fields such as healthcare, security, and robotics. In protecting SCADA (Supervisory Control and Data Acquisition) systems, CNNs could be used in several ways. For example, CNN could be trained to detect anomalies in the data collected by the SCADA system, such as sudden spikes in temperature or pressure. This could help identify potential system malfunctions, allowing operators to respond quickly to prevent more severe problems.

Additionally, a CNN could be used to monitor the network traffic of the SCADA system, looking for patterns or characteristics indicative of a cyberattack. This could help identify potential security threats in real time, allowing organizations to take appropriate action to protect the system. While CNNs are not a complete solution for protecting SCADA systems, they can be a valuable tool for detecting anomalies and identifying potential security threats.

LSTM-RNNs are a type of artificial neural network well-suited to tasks involving sequential data. They can retain information about previous inputs in the sequence and use it to make more accurate predictions or decisions. An LSTM-RNN could be trained on network traffic data from the SCADA system to identify typical communication patterns between different devices in the system. The network could then monitor current traffic and identify anomalies or suspicious communication patterns that may indicate a cyberattack.

Self-similarity refers to the property of a time series data in which its statistical properties remain constant across different time scales. The Hurst parameter is a numerical measure of self-similarity in time series data. In anomaly detection, the Hurst parameter can provide additional information about the underlying structure of the data and help distinguish between normal and abnormal behavior. For example, in time series data generated by a stable system, the Hurst parameter is expected to be close to a constant value, whereas, in the presence of an anomaly, the Hurst parameter may change significantly. Incorporating the Hurst parameter into an anomaly detection model can improve its ability to detect subtle and complex anomalies that may not be easily visible from the raw data.

In this thesis, we propose an enhanced DL model for cyber security in SCADA systems. The proposed model is based on a combination of CNNs and LSTM networks for anomaly detection. The critical contribution of this research is the addition of self-similarity, represented by the Hurst parameter, to the CNN-LSTM model. The Hurst parameter provides insight into the persistence of trends in the data and helps detect anomalies in time series data. By incorporating the Hurst parameter into our model, we aim to enhance its ability to detect subtle and complex anomalies in SCADA systems.

1.3 RESEARCH OBJECTIVES

The primary goal of this research is to improve the CNN-LSTM algorithm for more reliable and efficient SCADA IDSs. This can be accomplished by pursuing the sub-goals listed below.:

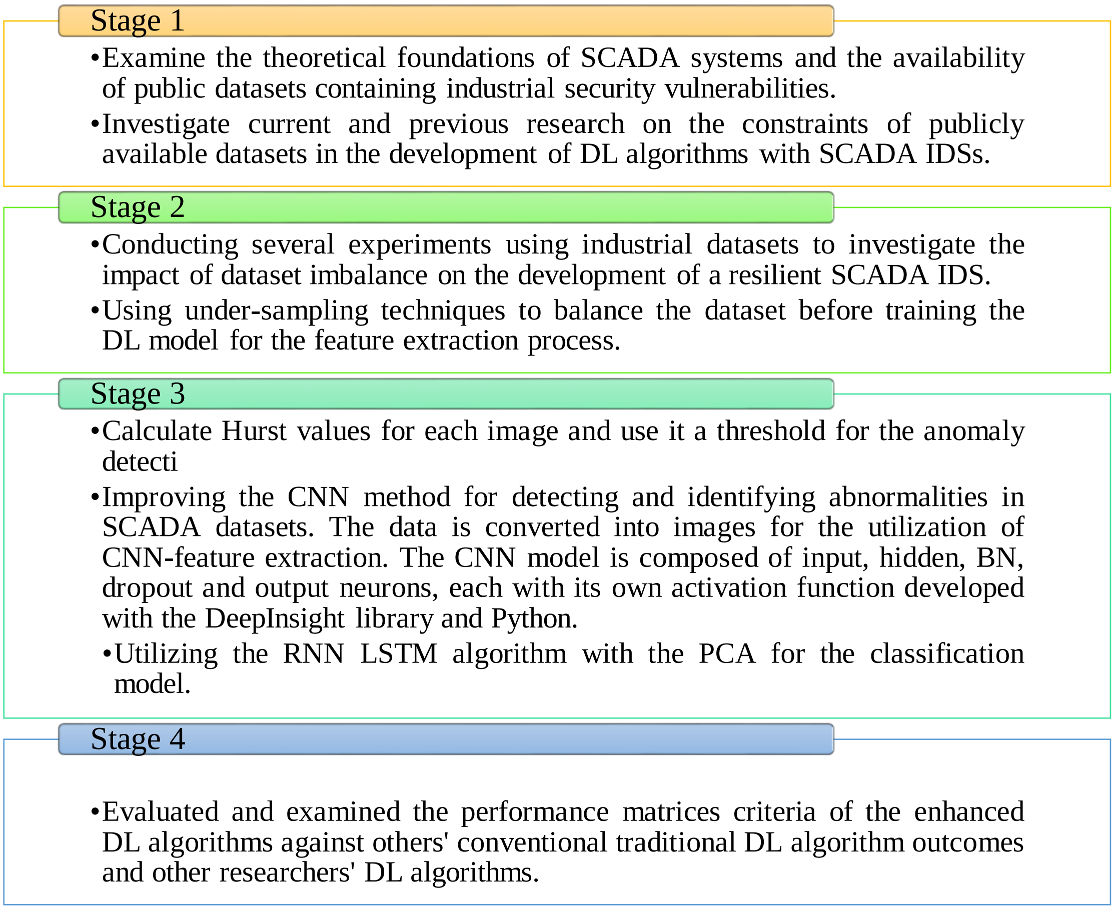
1. To investigate the suitability of existing public datasets for detecting and mitigating cyber attacks in SCADA systems.
2. To enhance DL algorithms with CNN and RNN LSTM neural network models by addressing the datasets imbalance issue, using BN, dropout layers, and the addition of self-similarity represented as Hurst parameter to detect and classify the cyber attacks in the SCADA datasets.
3. To evaluate and benchmark the performance of the proposed CNN-LSTM DL algorithm with balanced and imbalanced SCADA datasets.

1.4 RESEARCH METHODOLOGY

The below methods are followed in the thesis to achieve the mentioned objectives:

1. Examine the theoretical foundations of SCADA systems and the availability of public datasets containing industrial security vulnerabilities.
2. Investigate current and previous research on the constraints of publicly available datasets in developing DL algorithms with IDS.
3. Conducting several experiments using industrial datasets to investigate the impact of dataset imbalance on the development of a resilient SCADA Intrusion Detection System.
4. Using under-sampling techniques to balance the dataset before training the DL model for the feature extraction process.
5. Improving the CNN method for detecting and identifying abnormalities in SCADA datasets. The data is converted into images for the utilization of CNN feature extraction. The CNN model comprises input, hidden, BN to reduce the impact of changing input distributions during training, dropout layer to reduce the risk of overfitting, and output neurons, each with its activation function developed with the DeepInsight library and Python.
6. Calculation of Hurst parameter for each image.
7. We use the RNN LSTM algorithm with the Principal Component Analysis (PCA) for the classification model.
8. Evaluated and examined the performance matrix criteria of the enhanced DL algorithms against others' conventional traditional DL algorithm outcomes and other researchers' DL algorithms.

The research methodology stages are shown below in Figure. 1.2.



Stage 1

- Examine the theoretical foundations of SCADA systems and the availability of public datasets containing industrial security vulnerabilities.
- Investigate current and previous research on the constraints of publicly available datasets in the development of DL algorithms with SCADA IDSs.

Stage 2

- Conducting several experiments using industrial datasets to investigate the impact of dataset imbalance on the development of a resilient SCADA IDS.
- Using under-sampling techniques to balance the dataset before training the DL model for the feature extraction process.

Stage 3

- Calculate Hurst values for each image and use it a threshold for the anomaly detecti
- Improving the CNN method for detecting and identifying abnormalities in SCADA datasets. The data is converted into images for the utilization of CNN-feature extraction. The CNN model is composed of input, hidden, BN, dropout and output neurons, each with its own activation function developed with the DeepInsight library and Python.
- Utilizing the RNN LSTM algorithm with the PCA for the classification model.

Stage 4

- Evaluated and examined the performance matrices criteria of the enhanced DL algorithms against others' conventional traditional DL algorithm outcomes and other researchers' DL algorithms.

Figure 1.2 Research Methodology Stages

1.5 RESEARCH SCOPE

The research directions are restricted to the scope of this research listed below:

1. This research focuses on IDS in SCADA systems; other control systems are not discussed. This research may cover topics such as the challenges of developing IDS for SCADA systems, different types of IDS that can be used in SCADA, and the impact of dataset imbalance on the performance of these systems.
2. The CNN-LSTM method will be enhanced in the Python environment for intrusion detection rather than intrusion prevention.

3. The effectiveness of the DL algorithms presented in this research will likely be evaluated using performance measures such as accuracy, precision, sensitivity, and f1-score.

1.6 THESIS ORGANIZATION

This thesis is structured into five main chapters that detail the essential components of DL algorithms as they were built, developed, and evaluated during the research. These chapters begin with a summary of the field of research, an overview of SCADA and their applications, and the main reason to improve the CNN-LSTM algorithm. It also states the study's aims, problem statement, and scope of this research. The second chapter includes a brief history of SCADA systems and a discussion of different cyber attacks and DL/ML methods. Furthermore, it provides a theoretical examination of the DL/ML algorithms in intrusion detection for SCADA, as well as the characteristics of the techniques and the drawbacks of publicly accessible industrial datasets in the SCADA field. The output of Chapter Three is to define, discuss, and examine the research methodology. It also presents the DL parameters under study and explains the performance parameters. Chapter four discusses the implementation of the enhanced DL algorithms and thoroughly compares the study findings to previous publications in the field. Chapter Five discusses the future possibilities for DL algorithms, and conclusions are formed, see Figure. 1.3, which illustrates the thesis organization.

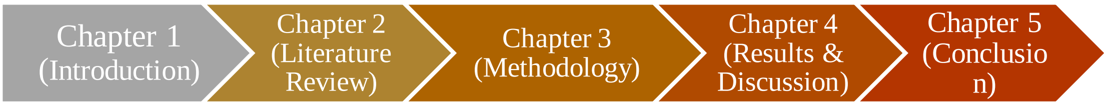


Figure 1.3 Thesis Organization

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

The Fourth Industrial Revolution, Industry 4.0, combines digital and physical technologies to allow responsive, networked operations. Businesses use AI, robotics, edge computing, and the cloud to make educated, timely decisions across the supply chain and intelligent manufacturing. Industrial Internet of Things (IIoT) solutions use connected sensors and edge devices in real time to improve product quality and overall operational efficiency. Many sectors use SCADA, including electric, water and wastewater, oil and natural gas, chemical, pharmaceutical, pulp and paper, food and beverage, and discrete manufacturing (automotive, aerospace, and durable goods) (Jamil, Ur Rahman, & Fawad, 2022). A SCADA system generally comprises control components (such as electrical, mechanical, hydraulic, and pneumatic) that work together to accomplish an industrial goal. The incorporation of IoT in SCADA systems improved communication between different components and provided a more efficient method of monitoring and data collecting. However, these improvements come at a cost: they break the isolation of SCADA systems, exposing them to cyber threats. This chapter provides a thorough introduction to the field of SCADA system security.

2.2 SCADA SYSTEMS

Data collection, communication systems, and Human Machine Interface (HMI) software are combined in the SCADA System to act as a centralized control and monitoring system for processing multiple inputs and outputs. The data obtained in the field is transmitted to the computer-based control center, where it is displayed textually or visually via HMI. SCADA systems are made up of both software and hardware. The hardware includes the control center's MTU. Communication facilities such as phone lines, radio, satellite, cable, and RTUs or PLCs are dispersed across sizeable geographical field locations to monitor sensors and operate actuators (Pliatsios, Sarigiannidis, Lagkas, & Sarigiannidis, 2020). The RTUs or PLCs control the local actions of the sensing devices, while the MTU analyzes and stores the data from the RTU's inputs and output. Information flows to and from the MTUs and RTUs via communication devices. The software defines when, what to monitor, the allowable parameter ranges, response style, etc.

2.3 SCADA VULNERABILITIES

More SCADA facilities are converting to an Internet Protocol (IP) architecture for wide-area communication, following the newest trend of employing standardized protocols. Standard protocols also increased system upgradeability by leveraging system implementation costs among providers. These developments, however, result in new threats and vulnerabilities. The increasing reliance on Internet-based communication has added to the issue's complexity and severity (Suaboot et al., 2020). Current SCADA networks are dispersed, networked, and rely on open internet protocols, making them vulnerable to global cyber-terrorism (H. Kim, 2012). There are mainly two categories of SCADA systems: a) the threat of unauthorized access to the control software and b) the risk that packets can access SCADA devices in host computers. The effects of a SCADA failure may be catastrophic and include financial loss due to machinery and environmental harm to the loss of life (Coffey et al., 2018).

SCADA systems are vulnerable to a range of security threats. These threats can come from various sources, including hackers, malware, and insider threats. Some common vulnerabilities of SCADA systems include:

- Outdated software and hardware: SCADA systems often use older software and hardware that may be vulnerable to security threats. These systems may not be able to receive regular security updates and patches, leaving them open to attack.
- Lack of encryption: Many SCADA systems do not use encryption to protect the data they transmit, making it easy for attackers to intercept and manipulate it.
- Weak authentication and access control: Many SCADA systems have weak authentication and access control measures, making it easy for attackers to gain unauthorized access to the system.
- Insufficient monitoring and detection: SCADA systems may not have adequate monitoring and detection capabilities, making it challenging to identify potential security threats promptly.
- Poor security practices: Many organizations that use SCADA systems may not have robust security practices in place, making it easy for attackers to exploit vulnerabilities in the system.

2.3.1 SCADA Cyber Attack Types

Cyber-physical systems protection is a concern for every nation globally due to the enormous number of electronic devices linked through communication networks. A few types of attacks that target SCADA systems are discussed below (Nasser et al., 2018).

- Worm: It is like viruses with no network guidance from attackers in their transmission. Unlike viruses, no intervention with the user is required in worms to enable their attempt to spread.
- Trojan: This is a type of software where disruptive functionality is applied to the original system.
- Virus: A virus is described as a code typically attached to another software, and while the program is running, it will run with it.
- DDoS: Coordinated assaults on the availability of a target device service or network implicitly launched via a series of corrupt computer devices.
- Targeted Attack: Refers to a malicious attack targeting a specific person, software, system, or organization. It could obtain data, interrupt activities, or delete data on the targeted device.
- Denial of Service: This attack is designed to prevent a network or a device from performing usual services. It is generally triggered when access to a network or computer resource is deliberately degraded or blocked because of another person's malicious behavior.

2.3.2 SCADA Latest Attacks

The launch of the first cyber weapon, identified as Stuxnet, marked a tipping point in the evolution of cybersecurity in June 2010. It targeted the Iranian nuclear facility at Natanz. Stuxnet was much more sophisticated than any other malware before. Still, it also took a different path that was no longer compatible with traditional confidentiality, integrity, and the principle of availability. More than 60,000 computers were infected by Stuxnet (Farwell & Rohozinski, 2011). The actual attack was not directed at SCADA software. Instead, it targeted the industrial controllers at the nuclear plant (Alladi, Chamola, & Zeadally, 2020). Likewise, in December 2014, hackers targeted a German steel plant, taking control of the production system and causing significant structural damage to the factory's manufacturing line (Lee, Assante, & Conway, 2014). The City of Atlanta, Georgia, and the Colorado Department of Transportation were targeted by SamSam ransomware in 2018. Cyber-attacks are often employed for reconnaissance and attacking critical infrastructure in the ongoing Russian-Ukrainian conflict.

2.4 SCADA-INTRUSIONS DETECTION SYSTEM

Through network traffic analysis, an IDS detects malicious attack activities. It strives to classify network packets as benign or malicious using rules, ML-based or DL-based models. It is successful, attracting the curiosity of many security researchers (Gu & Lu, 2021). An IDS can be deployed at the network's perimeter to safeguard all traffic entering the network; it is a hardware device known as a NIDS. Another typical device is the HIDS, a software IDS used to secure individual machines (see Figure 2.1). They are critical components of network security design (Mahdavifar & Ghorbani, 2019). The IDS might be signature-based, anomaly-based, or a combination.

Typically, the signature-based IDS is the standard for detecting SCADA attacks. It detects unique traffic data trends for identifying malicious activity, which can be applied as regulation rules in IDS applications such as Snort. One of the disadvantages of the signature-based approach is that it cannot detect zero-day attacks (C. Wang, Wang, Liu, & Qu, 2020). The anomaly-based approach creates a baseline for normal network behavior. Then, it identifies attacks by measuring the observed behavioral differences. Zero-day attacks can be discovered using the anomaly detection approach (Xu, Shen, Du, & Zhang, 2018). This is because it needs only normal operating conditions to learn the typical behavioral profile (Najafabadi et al., 2015). Several anomaly detection methods have been deliberately designed for ICS. However, despite significant advancements in IDS design, this field remains a work in progress (Hassan, Gumaei, Alsanad, Alrubaian, & Fortino, 2020). Data imbalance is one of the difficulties that security researchers face while constructing an ML/DL model. If the dataset used to train the SCADA IDS is imbalanced, it can negatively impact the performance of the IDS. For example, suppose the dataset contains many more examples of normal behavior than malicious behavior. In that case, the IDS may become biased toward recognizing normal behavior and may have difficulty identifying instances of malicious behavior. This can result in many false negatives (instances of malicious behavior that the IDS does not detect) and reduced overall system accuracy.

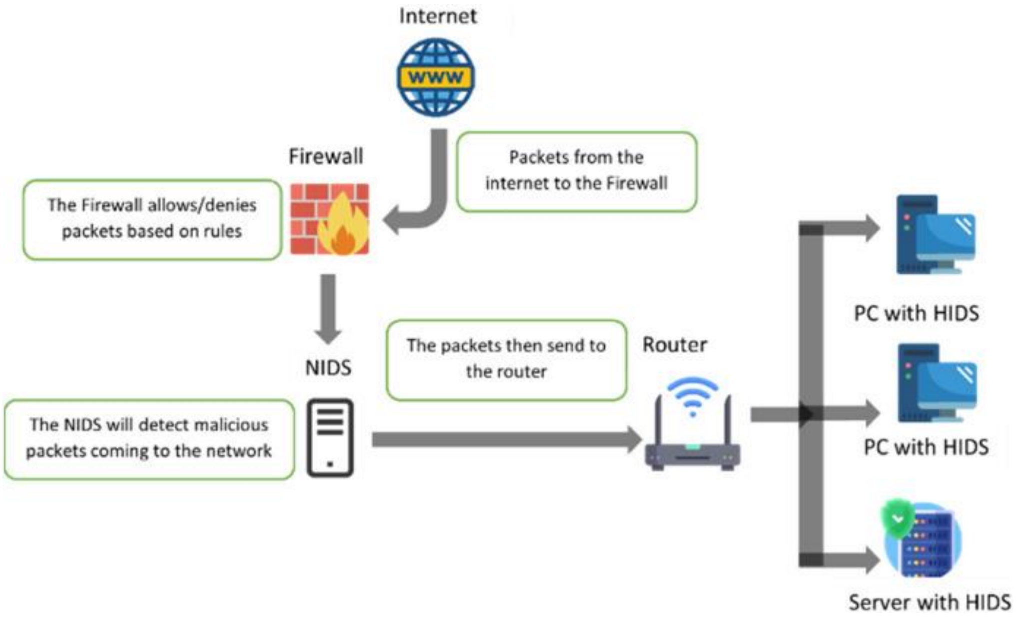

Internet
WWW
Firewall
Packets from the internet to the Firewall
The Firewall allows/denies packets based on rules
NIDS
The packets then send to the router
Router
The NIDS will detect malicious packets coming to the network
PC with HIDS
PC with HIDS
Server with HIDS

Figure 2.1 Network IDS (NIDS) vs. Host IDS (HIDS).

2.5 ML-BASED SCADA IDS TECHNIQUES

ML is a branch of AI, often overlapping with computational statistics, concentrating on making predictions. ML focuses on classification and regression based on the known features learned from training data. ML's pioneer, Arthur Samuel, described ML as a field of study that allows computers to learn without being explicitly programmed (Xin et al., 2018). ML methods are commonly used in many fields because of their computing power, and this section will discuss their applications. These methods focus primarily on classification, clustering, and regression based on previously learned characteristics from the training phase (Xu et al., 2018). To detect different attacks, ML can help the network administrator take effective action to avoid intrusion. Some of the most popular ML algorithms are the Support Vector Machine (SVM), K-nearest neighbor, and Decision Tree. SVM is an ML technique that has been applied for classification applications (see Figure 2.2), such as (Zolfi, Ghorbani, & Ahmadzadegan, 2019), where the authors constructed a model for cyber-crime classification (Aytug Onan & KorukoGlu, 2017)—presented a Naive Bayes and K-nearest neighbor model to classify texts (Abokifa, Haddad, Lo, & Biswas, 2019), designed an anomaly detection model to identify and classify attacks against cyber-physical systems. Unlike DL algorithms, which automate feature extraction, ML algorithms provide explicit steps to support a reached decision.

The primary issue with present ML (ML) algorithms is that they cannot extract the essential information from network packets for identifying the complicated nature of new attacks (Khraisat, Gondal, Vamplew, Kamruzzaman, & Alazab, 2019). Similarly, conventional feature extraction approaches, such as statistical and mathematical aspects, are insufficient for detecting intrusions that are ingeniously masked in plant data. In the SCADA networks, the amount of data generated is enormous, the IDS is expected to perform with low false alarms, and the percentage of malicious data is relatively small compared to the ordinary operation data (dataset imbalance). These challenges motivate the use of DL methods in developing IDSs.

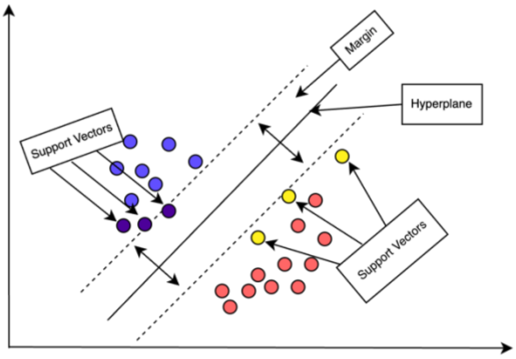


Figure 2.2 An example of ML Classification with the Support Vector Machine (SVM)

2.6 DL-BASED METHODS SIGNIFICANCE IN SCADA IDS

DL is an area of ML that simulates human brains for analytical learning. It mimics the human neural network mechanism for processing data such as images, sounds, and texts. It also has two learning methods: supervised and unsupervised. The essential characteristic that distinguishes DL from ML is the feature extraction process. DL solutions have achieved excellent results in various ML applications, including voice recognition, computer vision, and natural language processing (Najafabadi et al., 2015; Xu et al., 2018).

DL solutions have achieved excellent results in various applications, including voice recognition, computer vision, and natural language processing. The expanded use of DL techniques in the cyber security domain is the most extensive development, and researchers have recently identified the vast potential of deploying DL in the SCADA anomaly intrusion detection domain (Mulay, Devale, & Garje, 2010), Figure 2.3 provides the DL algorithms used in SCADA to detect intrusions.

The main difference between DL and ML is that a) DL requires a larger size of training data, b) the training time is longer in DL compared to ML methods, c) proper training in DL would benefit from a GPU, and d) the result is typically numerical in ML while DL results can be a voice, images, or numbers.

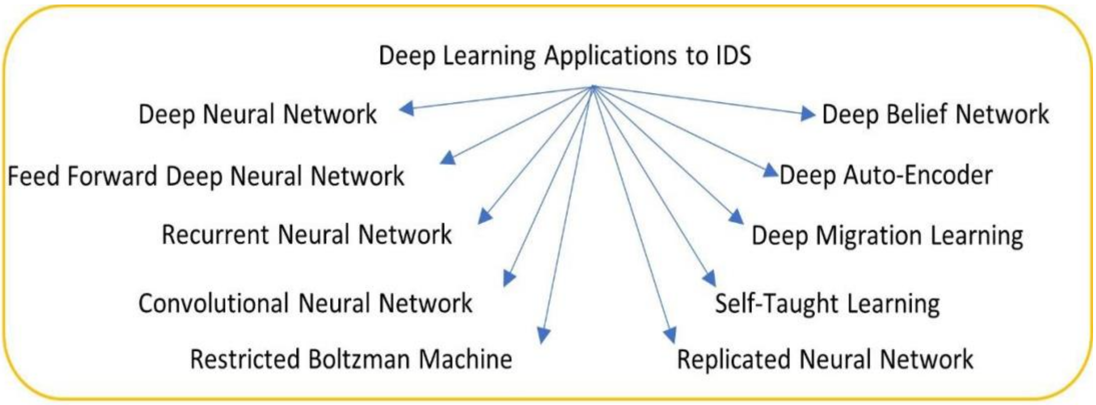

Deep Learning Applications to IDS
Deep Neural Network
Deep Belief Network
Feed Forward Deep Neural Network
Deep Auto-Encoder
Recurrent Neural Network
Deep Migration Learning
Convolutional Neural Network
Self-Taught Learning
Restricted Boltzman Machine
Replicated Neural Network

Figure 2.3 DL algorithms used in SCADA IDSs

2.7 DL ALGORITHMS – DBN, AUTOENCODER, CNN, AND RNN

Deep Belief Networks (DBNs) were created to address difficulties experienced while training deep-layered models with typical neural networks. These problems are the learning process is slow, manual parameter selection results in classification with the same local minima, and requires many training datasets. The DBN comprises multiple Restricted Boltzman Machines (RBMs); Figure 2.4 shows this architecture. An autoencoder is classified as an unsupervised learning neural network. It consists of three levels: input, output, and one or more hidden layers. Unlike other deep neural networks, an autoencoder provides a topology in which the hidden layers are smaller than the input layers. As a result, this can learn a precise characterization of features; Figure 2.5 shows the autoencoder design. CNN is a robust artificial intelligence algorithm. It has demonstrated exceptional feature extraction capabilities, notably in voice and image processing applications (Husaini, Habaebi, Hameed, Islam, & Gunawan, 2020). It is similar to a human neural network due to its weight-sharing arrangement, which minimizes the model's complexity and weight (Aleesa, Younis, Mohammed, & Sahar, 2021). The internal design is shown in Figure 2.6.

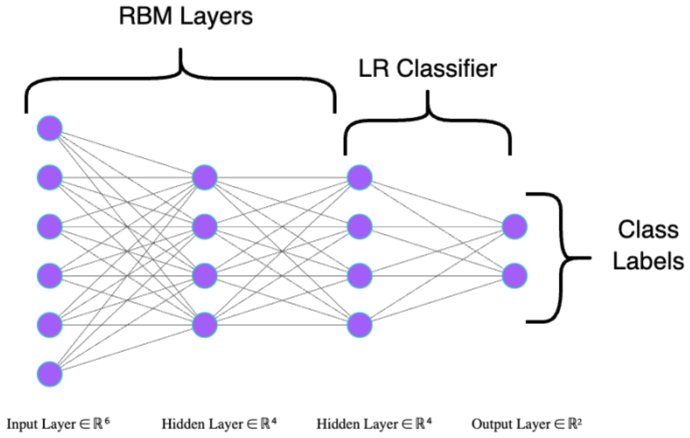


Figure 2.4 The internal architecture of DBN

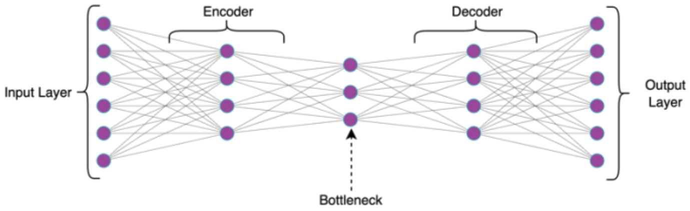


Figure 2.5 The design of autoencoders

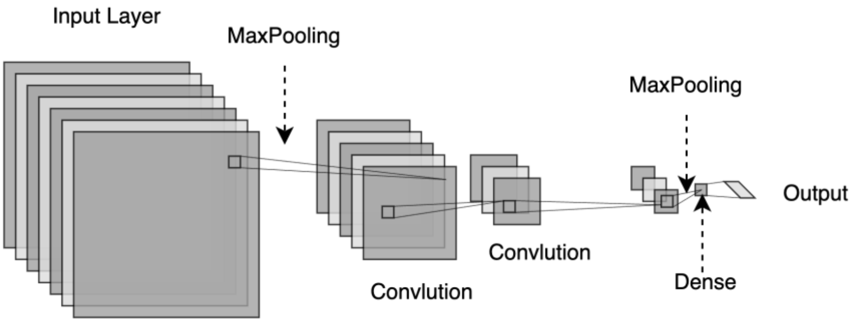


Figure 2.6 An example of the CNN network

The DL RNN technique can store and handle complicated data. RNNs are complex systems with an internal status at each classification point. This is because of circular relationships between neurons of higher and lower layers and optional self-feedback connections. These feedback relations enable RNNs to transmit data from previous events to current processing phases. Thus, RNNs construct a memory of the time series event (Staudemeyer, 2015).

The training process of the RNN classifier consists of two parts: Forward Propagation and Back Propagation. The feed-forward neural network enables transmitting information only in the forward direction, from the input nodes, through the hidden layers, and to the output nodes. There are no cycles or loops on the network. Decisions are based on the latest feedback in the feed-forward neural network. It does not memorize past details, and it does not have any potential scope. Feed-forward neural networks are used for general regression and classification problems. Back Propagation is necessary to send the residuals collected to change the weights, which are not inherently distinct from the ordinary neural network training (Yin, Zhu, Fei, & He, 2017); see Figure 2.7.

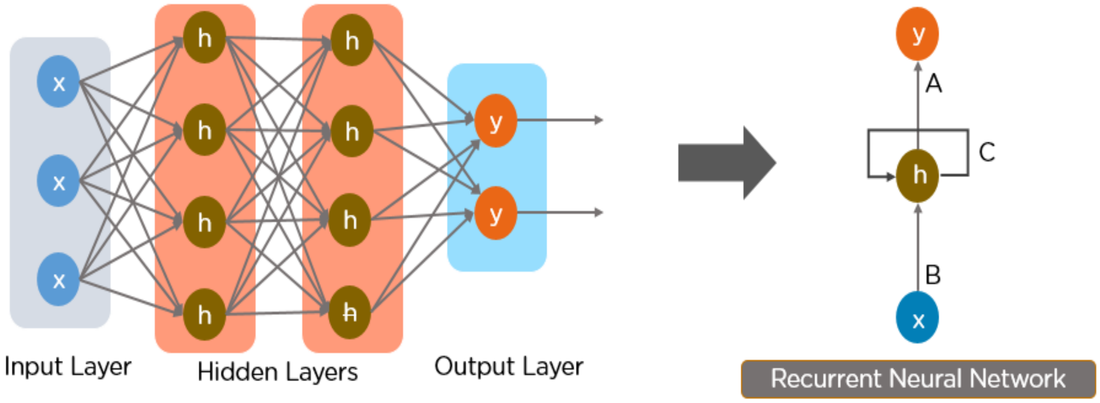


Figure 2.7 RNN model architecture ("Deep Learning with Tensorflow 2.0, Keras and Python | Codebasics," 2021)

2.7.1 DL Techniques- Related works

The review of the study of the implementation of the DL technique in IDS, along with its datasets, is highlighted in Table 2.1 below.

Table 2.1 DL Methods in SCADA IDS with Datasets

Author(s)	Dataset	Remarks	Limitations
(Tian et al., 2020)	NSL-KDD UNSW-NB15	A SCADA IDS Based on the enhanced DBN. The design focused on keeping the activation rate low to overcome the homogeneity of features issue. The sparsity penalty is used to overcome the overfitting in the dataset.	The NSL-KDD dataset is outdated, the training process is complex, and the feature extraction parameters are manually selected.
(Y. Li, Ma, & Jiao, 2015)	KDD99	DBN is used with Autoencoders to detect anomalies. AE is used to reduce the dimensionality in the dataset.	The KDD intrusion dataset is outdated. Attacks were related to the IT domain, and reducing the dimensionality of data may cause a loss of crucial features
(C. Wu et al., 2020)	NSL-KDD	Combined ML and DL methods (DBN-SVM) to detect intrusions in SCADA datasets. The Particle Swarm Optimization algorithm is used for optimization.	The dataset is outdated, with manual feature extraction.

(Marir et al., 2018)	KDD CUP′99 NSL-KDD UNSW-NB15 CICIDS2017	Four datasets are used to evaluate the SVM-DBN model.	KDD99 and the NSL are outdated datasets. It is not clear which dataset is used for the training. The evaluation metrics are not comprehensive.
(C. Wang et al., 2020)	SWAT	Autoencoder is used to reduce the data dimensionality in the training data.	The model can detect the device associated with the anomaly; it would be suitable for fault detection as it does not identify the attack type.
(Y. Yu, Long, & Cai, 2017)	CTU-UNB Contagio-CTU-UNB	Stacking dilated convolutional autoencoders to detect intrusions. The DCAE has fewer parameters than other DL models, so training takes less time.	Reducing the dimensionality of data may cause a loss of crucial features
(Zavrak & Iskefiyeli, 2020)	CICIDS2017	Implemented Variational Autoencoder (VAE) with One Class-SVM.	For manual feature extraction, the evaluation metrics are not comprehensive.
(G. Zhang et al., 2020)	NSL-KDD UNSW-NB15	Used Wasserste in Generative Adversarial Network (CWGAN) with Cost-Sensitive Stacked Autoencoders (CSSAE), with Autoencoders. Attempted to address the dataset imbalance by over-sampling the minority class.	The NSL-KDD dataset is outdated. Over-sampling may increase the training time and may lead to data corruption.

(K. Wu, Chen, & Li, 2018)	NSL-KDD	Developed a malware classification by transforming the data into pictures. The DL algorithm used in this work is CNN.	The NSL-KDD is an outdated dataset.
(W. Wang et al., 2017)	ISCX	CNN is used as a feature extractor. The paper showed the feasibility of using CNN to classify traffic data.	The time complexity analysis is not discussed, and the evaluation metrics are unclear.
(Khan, Zhang, Alazab, & Kumar, 2019)	KDD99	An enhanced CNN model for intrusion detection is introduced. It can automatically recognize features.	Outdated datasets, manual parameter selection, and evaluation are not comprehensive.
(Saxe & Berlin, 2017)	A raw string of data	CNN to detect malicious data in URLs, file paths, named pipes, named mutexes, and system files	small sample size, the time complexity is not discussed, and the evaluation is unclear.
(Staudemeyer, 2015)	KDD99	applied LSTM-RNN for the IDS implementation.	The outdated dataset is not performing well with Probe, R2L, and U2R attacks. Time complexity analysis is not discussed.
(J. Kim, Kim, Thu, & Kim, 2016)	KDD99 10 percent	Used LSTM-RNN algorithm for anomaly detection. The Hessian-Free algorithm is used for optimization.	Outdated dataset, high False Alarm Rate (FAR).
(Yin et al., 2017)	NSL-KDD	RNN-IDS is introduced for binary and multiclass classification.	An outdated dataset requires a lot of training time

(Kwon, Yoo, & Shon, 2020)	Raw traffic data.	Developed an IDS for IEEE 1815.1-based power system using Bidirectional-RNN. In bidirectional LSTM, the prediction is made by integrating forward propagation and backward propagation	The time complexity analysis is not discussed, and the evaluation process details of this model are not presented.
(Xu et al., 2018)	NSL-KDD	Deep Neural Networks (DBN) with Gated Recurrent Units (GRU) are combined to develop a SCADA IDS.	The outdated dataset and time complexity are not discussed.
(S. J. Yu, Koh, Kwon, Kim, & Kim, 2016)	KDD99	Using the Hurst parameter to detect anomalies in the network traffic.	Outdated dataset

The core points of the taxonomy analysis are summarized in Figure 2.8 and Figure 2.9. They summarize the algorithms used in their performance matrices and the datasets used to train and evaluate these models. To sum up the analysis, similarities and differences can be defined. A noticeable similarity is that ten papers used either the KDD99 dataset or a variation of the KDD99, like the NSL-KDD of KDD-10%. As a result, cyber-attacks can be categorized into four groups which are Denial of Service (DoS), User to Root (U2R), Remote to the user (R2L), and Normal. Another common issue is that the features are either chosen manually or extracted using an ML method instead of the DL algorithm. One of the main observed ideas is that they focused on reducing the dimensionality of the data.

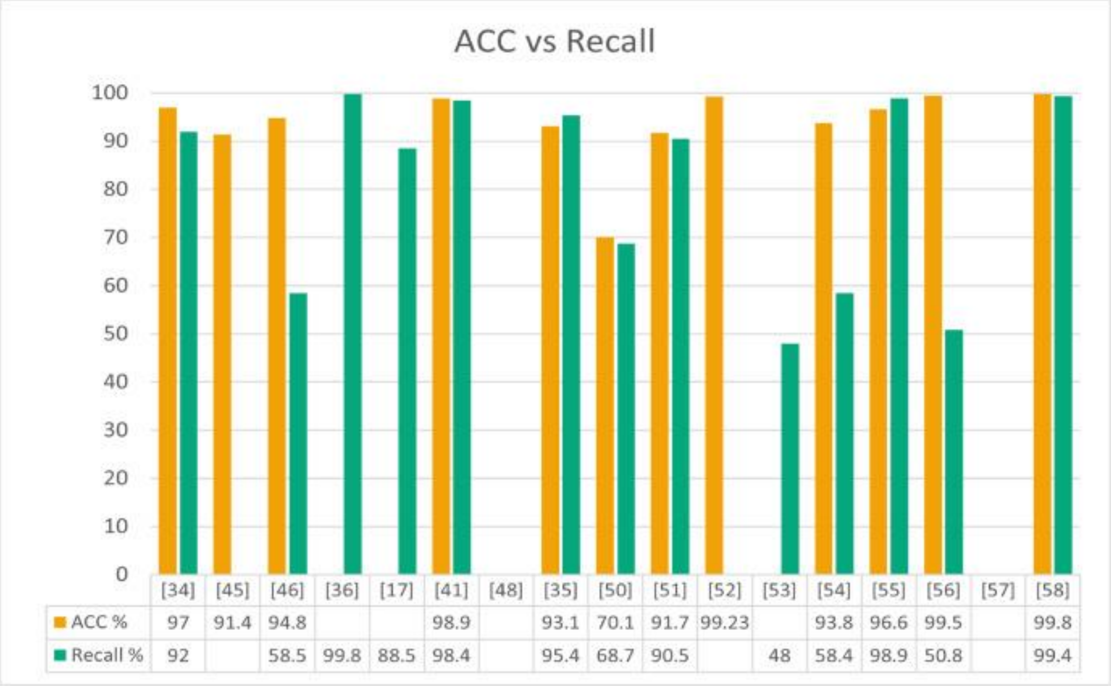


Figure 2.8 Accuracy and recall for different related works found in the literature.

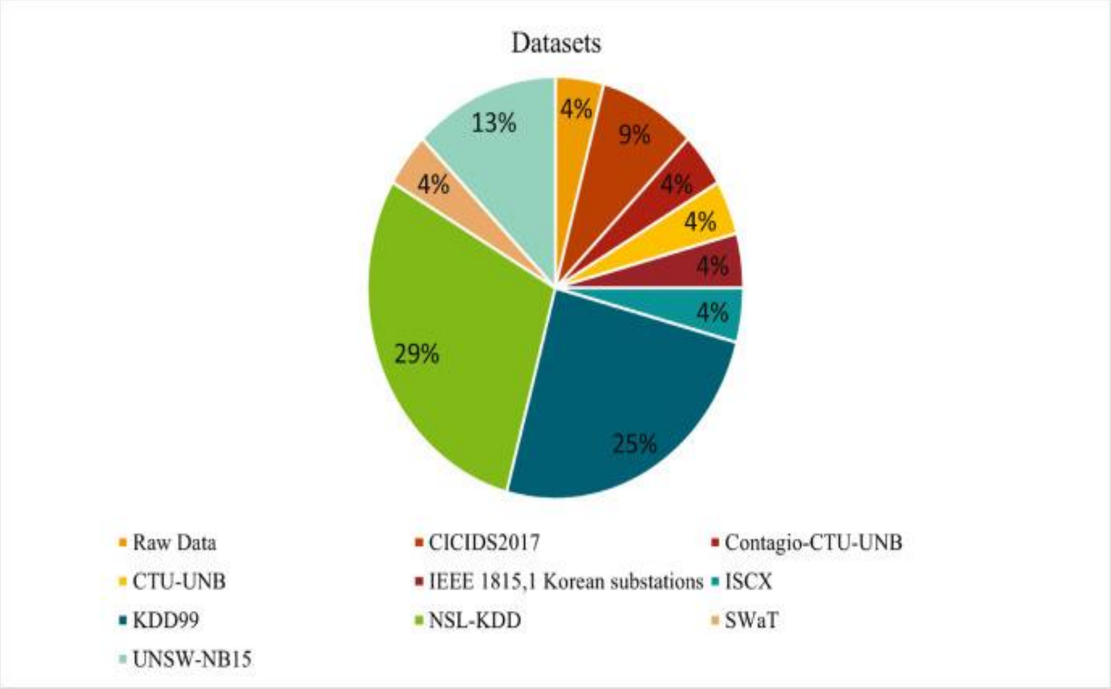


Figure 2.9 Datasets usage percentage in the different models discussed in section 2.7.1.

2.7.2 Limitations and Open Issues Found in the Literature

DL algorithms play an essential role in enhancing system security. Before implementing these algorithms into the SCADA network, several issues and problems should be addressed to increase network security. A spotlight on some of the issues and challenges is highlighted in this section.

Parameters tuning. The DL model parameters used in the literature are not specifically and carefully configured. DL parameters include the number of hidden layers, the activation function on each layer, the number of epochs, the optimization function, the learning rate, and the dropout rate. Predefined rules should be followed to have a reliable neural network architecture. For example, there are some formulas to determine the optimum number of hidden nodes in every layer. However, rules do not always lead us to the appropriate parameters and design. Some argue that all parameter configurations need to be assessed and the best design based on the best performance approaches to ensure the optimum parameter settings for a neural network.

Evaluation. There is no systematic method of selecting which DL model to use because each has its own characteristics that make it ideal for a particular application. All the papers produced an acceptable deep-learning method. Although each paper is evaluated with different metrics like accuracy, recall, and F1 score, the classification model's evaluation process is not standard. This makes it difficult to compare two approaches to the same problem. A straightforward procedure for the assessment of a DL classification algorithm is required.

Representation of cyber-physical threats. Most of the research on SCADA systems concentrated on cyber threats instead of physical and insider threats. SCADA becomes essential to people's lives; a cyberattack on the electric grid will directly affect people. In some cases, it may lead to the loss of lives. One of the fundamental issues that should be discussed in the future is isolating and mitigating an attack when it is discovered. The choice of the dataset is critical. Many researchers have used outdated datasets, such as the KDD99 group datasets. The use of such a dataset may lead to the issue of overfitting the model. The field of CPS security needs new and diverse datasets. Datasets can be generated from different ICT and CPS systems, such as additive manufacturing, water treatment plants, electricity generation, oil and gas plants, SCADA applications in consumer electronics, and many other fields. Such datasets could be extended to include IoT-based CPS of drones, self-driven vehicles, care area networks, drive-by-wire in airplanes, and mobile devices.

Data imbalance is a significant issue with ML and DL models. The labels in the dataset do not contain an equal percentage of normal and abnormal entries. In ML and DL applications, an imbalanced dataset is one in which the number of samples in one class is much lower than the number of entries in other classes, as discussed in (Mishra & Singh, 2021) and reaffirmed in (Ani, He, & Tiwari, 2017). Because of the data imbalance issue, the model may be biased toward a single class. There are a few ways to overcome this issue, either by under-sampling the majority class, as in the work of (Aytuğ Onan, 2019), or by oversampling the minority class (Mishra & Singh, 2021).

Graph-based security techniques. The field of network anomaly detection has made substantial use of graph theory. Several studies have used various graph-based techniques to detect anomalies (Ateş, Özdel, & Anarım, 2020). The graph properties were employed to take advantage of the spatial relationship in data transmission to detect botnets in the network (Chowdhury et al., 2017). (Pourhabibi, Ong, Kam, & Boo, 2020) reviewed various studies that utilized Graph-Based Anomaly Detection (GBAD) for fraud detection. To detect cyber-attacks against IoT networks, the authors (Abid &

Jemili, 2020) proposed a graph-based IDS. Graph-based IDS design is quite promising for detecting anomalies in SCADA systems.

Privacy by design (PbD) in SCADA. PbD is a concept that assures privacy protection by including privacy-enhancing technologies in the design standards of information technology, which makes privacy the default (Cavoukian, Polonetsky, & Wolf, 2010). PbD was utilized by researchers (O'Connor, Rowan, Lynch, & Heavin, 2017) to protect health information obtained through IoT devices. In (Pedraza, Patricio, de Asís, & Molina, 2013), the researchers offered another example of using PbD to develop an ML system. Although these principles were used to develop a Face-Recognition System, they can be replicated for SCADA IDS designs to offer plenty of security features.

Applications of Blockchain in SCADA security. Because SCADA systems are increasingly adopting IIoT, Blockchain technologies can be used to protect such networks. The utilization of Blockchain to develop IDSs in IIoT systems was demonstrated by (Derhab et al., 2019). Similarly, (Vargas, Lozano-Garzon, Montoya, & Donoso, 2021), the authors combined ML methods and Blockchain techniques to transfer information between sub-networks securely. The work by (Alladi, Chamola, Rodrigues, & Kozlov, 2019) reviews different Blockchain applications in smart grid security.

2.8 SELF-SIMILARITY FOR ANOMALY DETECTION

Self-similarity is often used in anomaly detection to identify deviations from the expected patterns in time series data. The idea is to compare the statistical properties of the time series data at different time scales and identify any changes in these properties as potential anomalies. The Hurst parameter is a standard measure of self-similarity in time series data, and it provides information about the persistence of trends in the data. The Hurst parameter indicates a random process if it is close to 0.5. In contrast, values greater than 0.5 indicate persistence (trends tend to persist over time), and values less than 0.5 indicate anti-persistence (trends tend to reverse over time). In a stable system, the Hurst parameter is expected to be close to a constant value, whereas, in the presence of an anomaly, the Hurst parameter may change significantly. By using self-similarity, especially the Hurst parameter, as a feature in an anomaly detection model, one can enhance the model's ability to detect subtle and complex anomalies that may not be easily visible from the raw data. This is because the self-similarity feature provides additional information about the underlying structure of the data, which can help distinguish between normal and abnormal behavior.

2.9 PUBLIC SCADA DATASETS

Data is the foundation for security analysis in SCADA IDS, and this section presents the standard public datasets in SCADA. As shown in section 2.7, most studies examine DL detection skills utilizing publicly accessible SCADA datasets. Most datasets are out of date and only relevant to IT systems. KDD99 by Hettich in 1999 ("KDD Cup 1999 Data," 2021) contains around 5,000,000 records, each of which includes 41 attributes and is labeled either as regular or as an attack, with one unique form of attack. The NSL-KDD (Tavallaee, Bagheri, Lu, & A. Ghorbani, 2009) is a variation of KDD99 created in 2009, and it has some advantages compared to KDD99. For example, the redundant record is excluded from the train collection to reduce bias toward the most common records, and the duplicate record in the test sets is removed. They are gathered by MIT Lincoln's Cyber Systems and Technologies (Lippmann et al., 1999). Another dataset used in SCADA IDS development is ISCX2012 (Shiravi, Shiravi, Tavallaee, &

Ghorbani, 2012). The data was gathered at the University of New Brunswick. It was gleaned through network traffic, which included HTTP, SMTP, SSH, and FTP, among other protocols (Mohamed, Dahl, & Hinton, 2010).

The UNSW-NB15 dataset was created by the University of New South Wales' cybersecurity laboratory group. Several researchers have used this dataset to create a SCADA IDS, such as those in the work of (Tian et al., 2020), in which the authors developed an IDS model based on a deep belief network (DBN) (Marir et al., 2018). combined ML techniques with DL algorithms to detect abnormality in network traffic (H. Zhang, Huang, Wu, & Li, 2020). used this dataset to improve the detection of minority and zero-day attacks on the networks.

The CICIDS2017 (Canadian Institute for Cybersecurity IDSs) dataset is another dataset used in SCADA systems, including modern and known recent attacks (Sharafaldin, Habibi Lashkari, & Ghorbani, 2019). It consists of raw data and traffic records collected over five days.

There are two SCADA datasets used in Water Distribution Systems (WDS). One of the new SCADA datasets is the BATtle of the Attack Detection Algorithms (BATADAL) (Taormina et al., 2018). Training Dataset 1, Training Dataset 2, and Test Dataset are the three datasets that comprise this dataset. The Secure Water Treatment (SWaT) (Goh, Adept, Junejo, & Mathur, 2017) dataset was generated for cyber-security research at the Singapore University of Technology and Design. Data is marked by normal and abnormal behavior (Inoue, Yamagata, Chen, Poskitt, & Sun, 2017).

The CTU-UNB dataset combines data from the CTU-13 dataset with normal records from the UNB ISCXIDS 2012 dataset (Y. Yu et al., 2017). Several researchers use this dataset to assess the implemented SCADA system's security measures. The Power System Attack (C et al., 2014) datasets consist of three datasets: a) 2 classes, b)

3 classes, and c) multiclass. Oak Ridge National Laboratories and Mississippi State University created them. These datasets have been implemented in many cyber-physical systems for the smart grid (Pan, Morris, & Adhikari, 2015). Table 1 provides a summary of the public datasets utilized in SCADA IDS. Many researchers are now studying DL/ML detection algorithms using publicly available SCADA datasets.

Table 2.2 Public SCADA datasets

SCADA Dataset	Year	Availability	Remarks
KDDCup 1999	1999	(“KDD Cup 1999 Data,” 2021)	widely used SCADA systems.
NSL-KDD	2009	(Choudhary & Kesswani, 2020)	A better version of KDDCup99.
ISCX Dataset	2012	(Shiravi et al., 2012)	Used for anomaly detection on the network.
UNSW-NB15 Dataset	2015	(Moustafa, 2021)	has 49 features and a collection of regular and attacked events, with around 2.5 million class records.
CICIDS2017	2017	(Sharafaldin et al., 2019)	A date, source and destination IPs, source and destination ports, protocols, and attack type are all included in each entry.
BATADAL Datasets	2018	(Taormina et al., 2018)	Used in Water Distribution Systems.

SWaT Dataset	2017	(Goh et al., 2017)	Used in Water Distribution Systems.
CTU-UNB Datasets	2013	("The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background Traffic. — Stratosphere IPS," n.d.)	It includes attack types such as Web-based malware, Exploits, and Botnet.
Power System Attack Datasets	2014	(C et al., 2014)	Used to detect intrusions in the smart grid.
Morris Gas Pipeline	2013	("Tommy Morris - Industrial Control System (ICS) Cyber Attack Datasets," n.d.)	A corpus of marked RTU telematics streams generated by a gas pipeline system
Morris Power System	2014	("Tommy Morris - Industrial Control System (ICS) Cyber Attack Datasets," n.d.)	The data is a combination of field device measurement and device logs. It consists of three datasets: a) binary dataset, b) three-class dataset, and c) multiclass dataset.
Bot-IoT (5%)	2018	("The Bot-IoT Dataset | UNSW Research," n.d.)	Bot-IoT was built on a testbed with different virtual computers running different operating systems.
ICS cyber testbed industrial OT dataset	2022	(Mubarak et al., 2022)	It was developed from a portable ICS testbed that included a) a PLC system, b) an HMI system, c) an Ethernet switch, d) Process simulation modules, e) a Physical Sensor, and f) an Attacker system.

2.10 DATASET BALANCING TECHNIQUES

This section highlights the importance of balancing datasets and the techniques used. SCADA systems monitor and control industrial processes in power plants or factories. These systems typically generate large amounts of data, which can be used for various purposes, such as to improve the efficiency of industrial processes or to detect anomalies that may indicate potential problems.

However, the data generated by SCADA systems are often imbalanced, with some classes or categories being underrepresented. This can be a problem when training DL models on the data, as the models may be biased towards the majority class and not accurately capture the minority classes. The dataset should be balanced using sampling approaches to solve this problem. Sampling approaches select a portion of the data to train the model to ensure that all classes are represented equally. There are several ways to do this, including under-sampling the majority class and oversampling the minority class (A. Chawla, Lee, Fallon, & Jacob, 2019).

Under-sampling implies lowering the number of samples in the majority class to equal that in the minority class. This can be accomplished by random sampling, which selects a random subset of the majority class, or through targeted sampling, which selects samples based on specific criteria, such as distance to the classifier's decision boundary.

Conversely, oversampling involves increasing the number of samples in the minority class to match the number in the majority class. This can be done through various methods, such as generating synthetic or repeating samples from the minority class.

2.10.1 Under-Sampling Approaches

Under-sampling is a popular technique for balancing SCADA datasets. This approach involves reducing the number of samples in the majority class to match the number in the minority class. Random sampling involves selecting a random subset of the majority class to reduce its size (Miah, Khan, Shatabda, & Md.Farid Dewan, 2019). This approach can be simple and effective but may not always produce the best results.

Targeted sampling, conversely, involves selecting samples from the majority class based on specific criteria. For example, samples closest to the classifier's decision boundary may be selected, as they are likely to impact the model's performance most.

Another approach to under-sampling is to use a clustering algorithm to group similar samples together and then select a representative subset of each cluster to reduce the overall size of the majority class (Aziz & Ahmad, 2021). This can help preserve the majority class's diversity while reducing its size to match the minority class.

Under-sampling can be a valuable technique for balancing SCADA datasets, as it allows for developing more accurate and fairer machine-learning models. However, it is crucial to carefully select the samples to be removed to avoid losing important information that may be useful for training the model (Tsai, Lin, Hu, & Yao, 2019).

Near Miss sampling is a method that involves selecting a subset of the majority class that is like the minority class. This is done by selecting observations from the majority class nearest to the classifier's decision boundary. The selected observations are removed from the dataset, resulting in a more balanced dataset.

Condensed under-sampling is a method that involves creating a new, smaller dataset by selecting a subset of the majority class that is most representative of the overall distribution of the classifier. This is done by selecting a subset of the majority class that contains a similar number of observations to the minority class but also maintains the overall distribution of the classifier.

Tomek link is a method that removes observations from the dataset between two observations of different classes. This is done by identifying pairs of observations nearest neighbors of opposite classes and removing the observation between the two. This results in a more balanced dataset.

The Edited Nearest Neighbors rule is a method that involves removing observations from the dataset that are not representative of their class. This is done by identifying the k-nearest neighbors of each observation and removing the observation if most of its neighbors belong to a different class.

One-sided selection is a hybrid method that involves first applying over-sampling to the minority class and then applying under-sampling to the majority class. This results in a more balanced dataset that contains a larger number of observations of the minority class and a smaller number of observations of the majority class.

2.10.2 Over-Sampling Approaches

Over-sampling is a technique used in ML to address the class imbalance in a dataset. Class imbalance occurs when the number of observations from one class far outnumbers the observations from the other classes. Over-sampling involves increasing the number of observations of the minority class in the dataset to better balance the classes (Mahmoud, El-Kilany, Ali, & Mazen, 2021).

There are several different approaches to over-sampling. One of the most common is random oversampling, duplicating observations from the minority class to balance the dataset. Another approach is called synthetic oversampling, which involves generating new observations for the minority class using a combination of the existing observations.

One of the advantages of over-sampling is that it can be performed on a dataset without requiring additional data. This makes it a simple and effective way to address the class imbalance. However, over-sampling can also introduce bias into the dataset if it is not done carefully. It is important to ensure that the new observations added to the dataset represent the minority class and not simply duplicates of existing observations (N. v Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

Random oversampling involves increasing the number of observations of the minority class in the dataset to better balance the classes. This is done by simply duplicating observations from the minority class.

The Synthetic Minority Oversampling Technique (SMOTE) is a popular method that combines random and synthetic oversampling to create new observations for the minority class (N. v Chawla et al., 2002). This is done by selecting observations from the minority class and using them to generate new synthetic observations. The new synthetic observations added to the dataset is a popular method that combines random and synthetic oversampling to create new observations for the minority class. This is done by selecting observations from the minority class and using them to generate new synthetic observations.

Adaptive Synthetic Sampling (ADASYN) is an oversampling method that combines the existing observations and the borderline samples of the classifier to generate new observations for the minority class. Borderline samples are observations located near the classifier's decision boundary and are essential for determining the classification of new observations.

Table 2.3 Advantages and Disadvantages of Under-Sampling and Over-Sampling.

Technique	Advantages	Limitations
Under-sampling	Reduces bias towards majority class, simple and effective approach	Loss of essential features potential imbalance, if not done carefully, requires sufficient data
Over-sampling	Reduces bias toward the majority class. No additional data is required	It may cause imbalance towards minority classes, increase dataset size, and make training difficult

Balancing a dataset for IDSs (IDS) using under-sampling and oversampling is a common problem in ML. Class imbalance occurs when the number of observations from one class far outnumbers the observations from the other. Under-sampling is a technique used to address the class imbalance by reducing the number of observations of the majority class in the dataset. Oversampling is a technique used to address the class imbalance by increasing the number of observations of the minority class in the dataset. Balancing a dataset for IDS using under-sampling and oversampling can help improve an ML model's performance by providing a more balanced dataset for training. It is essential to carefully evaluate the methods used to balance the dataset and ensure they do not introduce bias or cause vital information to be lost.

2.11 SUMMARY

This chapter focuses on the theories and terminology underlying this study, such as SCADA systems and applications, SCADA security open issues, IDSs, advanced DL DBN, Autoencoders, CNN, and RNN approaches, and their importance in developing a reliable, resilient, and effective SCADA IDS. The capabilities of intrusion detection and its limitations using public datasets, as well as a comparison of DL algorithms, are discussed in detail in this chapter. In addition, the existing public industrial datasets were studied, and the approach for an enhanced IDS by addressing dataset imbalances is followed in the next chapter.

CHAPTER THREE

METHODOLOGY

3.1 INTRODUCTION

This chapter explains the research methods and procedures for achieving the research questions. A thorough screening of the public SCADA datasets is introduced. The research starts with a literature review of the current state of the art. The datasets used to develop IDSs and the limitations. SCADA systems control and monitor critical infrastructure and industrial processes, such as power plants and manufacturing facilities. These systems are essential for the smooth operation of many industries. If they were to be compromised, they could have serious consequences, such as disruptions to essential services, financial losses, and even threats to public safety. Therefore, it is essential to evaluate the security measures for SCADA systems to ensure they are adequately protected against potential security threats. This can involve a range of activities, such as conducting risk assessments, implementing security controls, and regularly testing the effectiveness of security measures. By taking these steps, organizations can help to ensure the continued safe and secure operation of their SCADA systems.

This work addresses the dataset imbalance and improves the CNN-LSTM algorithm for more reliable and efficient SCADA IDSs. The specific goals are to screen publicly available SCADA datasets, conduct experiments to understand the impact of dataset imbalance on SCADA IDS development and evaluate the enhanced CNN-LSTM algorithm against other works.

This chapter is organized as follows: Section 3.2 outlines the materials and techniques used to screen the SCADA datasets by providing a clear guideline for the dataset criteria, data collection, and how to evaluate a SCADA dataset. Section 3.3 describes the experiment parameters and tools used to investigate the impact of the dataset imbalance problem. Section 3.4 explains how the findings are examined, and Section 3.5 summarizes the chapter.

3.2 DATASET SCREENING

SCADA datasets are data collections used to train and evaluate ML algorithms for SCADA IDSs. These datasets typically include sensor readings, control signals, and other data collected from SCADA systems. They may be collected in various industrial environments, including power plants, manufacturing facilities, and water treatment plants. This section provides how we examined the public datasets associated with SCADA to improve IDSs.

In the literature review, significant databases such as Elsevier, IEEE Explore, Microsoft Academic, Springer, Google Scholar, and Wiley online libraries are utilized to search for papers relevant to IDSs for SCADA. The methods and standards adopted to include a scientific paper in this study focus on answering SCADA intrusion detection problems by applying DL techniques. This research was limited to articles published between 2015 and 2022.

The outcome of the literature review provided a highlight for the SCADA datasets used by security researchers to develop IDSs. The following datasets are found in the literature; 1) UNSW-NB15, 2) CICIDS2017, 3) KDD99, 4) NSL-KDD, 5) ISCX2012, 6) Morris Gas Pipeline, 7) Morris Power System, 8) ICS cyber testbed industrial OT dataset, 9) BATADAL Datasets, 10) SWaT Dataset, 11) CTU-UNB Datasets, 12) Power System Attack Datasets.

The dataset examinations are based on multiple factors. Including a) how a dataset is cleaned, trained, and evaluated in IDSs, b) the results of using the dataset in developing IDSs for SCADA, and c) the drawbacks of a dataset and how it affects the development of IDSs. The constraints and issues associated with using such datasets.

Each dataset is analyzed using the EDA method to gain a comprehensive overview of the attributes, size, and attack types. The EDA's primary goal is to uncover inconsistencies in the dataset, recognize common patterns, and identify anomalies. It enables us to fully understand before forming any assumptions or hypotheses. EDA involves using various statistical and visualization techniques to explore and analyze the data and can help uncover hidden relationships and patterns that may not be immediately apparent. This can give vital insights into the data's characteristics and help identify the most relevant attributes the IDS should be trained with. The following section will explain the process of performing EDA with Python.

3.1.1 Performing EDA

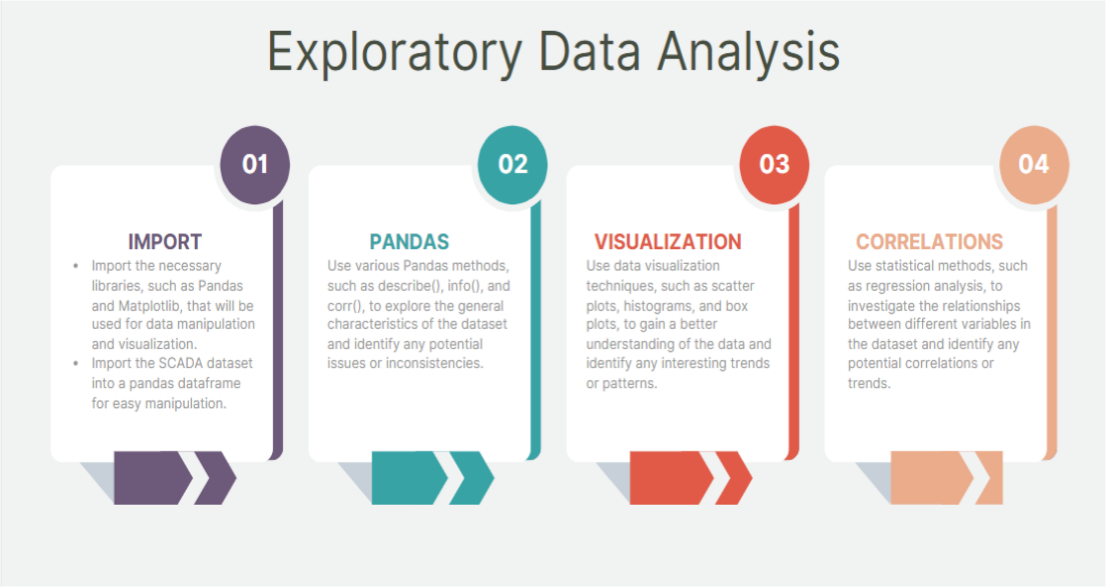
Figure 3.1 depicts the EDA workflow in Python. The following is a list of all the steps required to conduct an effective and meaningful EDA on a dataset.

1) To begin, load the Python libraries that will be used in the analysis. `Pandas`, `Numpy`, `Matplotlib`, `Seaborn`, and `Scatter` are among these libraries. The dataset is loaded into a Pandas data frame, which we'll refer to as `df`.
2) Using Pandas function `info()` will give us basic information about the data, such as the columns, number of records, null data, and the data type. Then, we can use another function in the Pandas library, `describe()`. This function will provide a statistical overview of each column, the number of records in each

column, the average, maximum value, minimum value, and standard deviation. We can continue exploring the data with functions like `duplicated()` and `unique()`.

3) Visualize the data using libraries such as `Matplotlib`, `Seaborn`, and `Scatter`. Visualizing the data can help us identify inconsistencies, patterns, and anomalies.

4) It is to find the correlated variables and features in the dataset using the `corr()` function and visualize the data with `Seaborn`. Overall, EDA is an essential step in developing an IDS, and using Python can provide a powerful and flexible platform for exploring and analyzing SCADA datasets.



- Import the necessary libraries, such as Pandas and Matplotlib, that will be used for data manipulation and visualization.
- Import the SCADA dataset into a pandas dataframe for easy manipulation.

Use various Pandas methods, such as describe(), info(), and corr(), to explore the general characteristics of the dataset and identify any potential issues or inconsistencies.

Use data visualization techniques, such as scatter plots, histograms, and box plots, to gain a better understanding of the data and identify any interesting trends or patterns.

Use statistical methods, such as regression analysis, to investigate the relationships between different variables in the dataset and identify any potential correlations or trends.

Figure 3.1 Dataset EDA workflow.

3.2.1 Dataset Cleaning

After performing EDA on SCADA datasets, the next step is to clean and preprocess the data. This involves removing outliers or missing values and transforming the data into a format suitable for further analysis. It may also be necessary to perform feature engineering, which involves creating new features from existing data to help improve DL models' performance. Generally, once the data has been cleaned and preprocessed, the next step is to apply DL algorithms to the data to build predictive models or identify patterns and trends. In our case, we investigated the impact of the dataset's imbalance. Hence, the dataset should be balanced before being used as training data for the DL model. A balanced dataset contains approximately 50% of each majority and minority class.

The classification of public SCADA datasets is the result of the datasets screening. The outcome is the characteristics of each dataset. These criteria include the data type, whether simulated data, real-world SCADA field data, network traffic data, or log files. Other properties are the dataset format, protocols found in the records, number of features, class distributions, size, and the Attack to Normal Ratio (ANR); more details in Chapter 4. The following section will use the imbalanced datasets to understand the impact of the dataset imbalance in developing IDSs for SCADA systems.

3.2.2 PCA

The PCA method is a statistical methodology for analyzing the variability of a multivariate dataset. PCA is used in the context of SCADA data to uncover patterns and trends that may not be immediately apparent, as well as to reduce the dimensionality of the data by projecting it onto a lower-dimensional space. This may be used for various purposes, such as data visualization, anomaly detection, and feature selection. In this scenario, PCA is utilized to comprehend and gain a reasonable dataset overview. To get an idea of a city, we do not need to see every street; just a few would work. Similarly, for reducing the dimensionality of a dataset to understand its general properties. To perform the PCA on a SCADA dataset in Python, you may follow the next steps:

1. Import PCA from the sklearn library.
2. Load the dataset.
3. Standardize the data by subtracting the mean and dividing it by the standard deviation.
4. Fit the PCA model to the data
5. Transform the data into the principal components

The first step is to load the SCADA dataset and standardize it by subtracting the mean from each value and dividing it by the standard deviation. Then, the `fit()` method creates a PCA instance and fits the data. Finally, the `transform()` method transforms the data into the principal components.

3.3 EXPERIMENT SETTINGS

The primary goal of this section is to understand the impact of dataset imbalance in SCADA IDS by conducting a few experiments. This section uses two imbalanced datasets: the Morris Power Dataset and the CICIDS2017 dataset. The Google Collab platform runs the Python commands for its ease of use and provides GPU access to improve the model's training. The model used for this purpose is the CNN-LSTM.

Four experiments were conducted to determine the effect of dataset imbalance. In the first one, CNN-LSTM detects intrusions using imbalanced data. In the second experiment, the data is balanced using under-sampling only. The model is trained with balanced data using an oversampling approach in the third experiment. A hybrid balancing technique is used in the fourth experiment, under-sampling the majority class and over-sampling the minority class. Next, the CNN-LSTM model is used to detect anomalies in the dataset. Each experiment's average values have been reported after being conducted several times. The DL model was built with the TensorFlow, Pandas, and Keras frameworks. The measures we used to assess the performance of these experiments are described next.

3.3.1 Evaluation Metrics

The evaluation metrics used in these experiments are briefly discussed in this section. All experiments evaluate the model based on Accuracy (ACC), Recall, Precision, and F1-score. Accurate is the most common performance metric for binary and multiclass classification problems. An IDS accuracy rate measures how accurately it detects normal or abnormal network traffic (Z. Wang, Xie, Wang, Tao, & Wang, 2021). The TPR is the ratio of correctly predicted and total network anomalies. TPR is called Recall or sensitivity. The Precision rate is an indicator of accuracy, which indicates the proportion of the number of positive cases correctly classified by the classifier to the number of positive cases. F1-score is the weighted harmonic average of Precision and Recall, which is quite effective for the imbalanced classification problem.

3.3.2 Experiment 1 - CNN-LSTM with imbalanced datasets

The imbalanced SCADA datasets are adjusted using the MinMaxScaler, with 70% of the data used for training and 30% for testing. Figure 3.2 shows the flow of this experiment. This experiment consists of three steps, which are as follows:

1. Preprocessing of dataset. Categorical features are converted to numerical features in this step. The data values are then normalized between 0 and 1 to expedite the transformation process. Furthermore, any instance with missing values is removed, as is any feature with the same value for more than 80% of all records.

2. Training and Testing. The CNN-LSTM model is enhanced with BN and layer modifications, and the best parameters for training the dataset are chosen.
3. Evaluation stage. Accuracy, Recall, and F1-score metrics evaluate the model's performance.

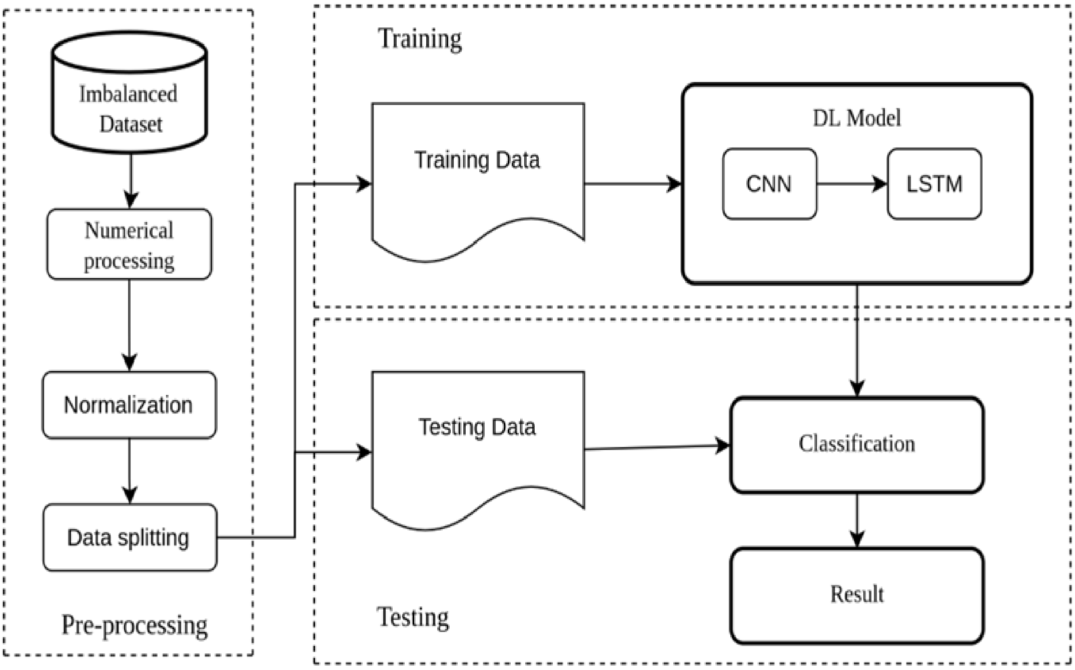


Figure 3.2 The flowchart of training the CNN-LSTM model with imbalanced data.

3.3.3 Experiments 2,3 & 4 - CNN-LSTM with balanced datasets

In experiments 2,3 and 4, the dataset is divided based on its majority and minority classes. Experiment 2 trains the model with a dataset balanced with under-sampling approaches. In experiment 3, the datasets are balanced with over-sampling techniques. For experiment 4, both under-sampling and over-sampling techniques are used. Figure 3.3 illustrates the workflow used in each of these experiments.

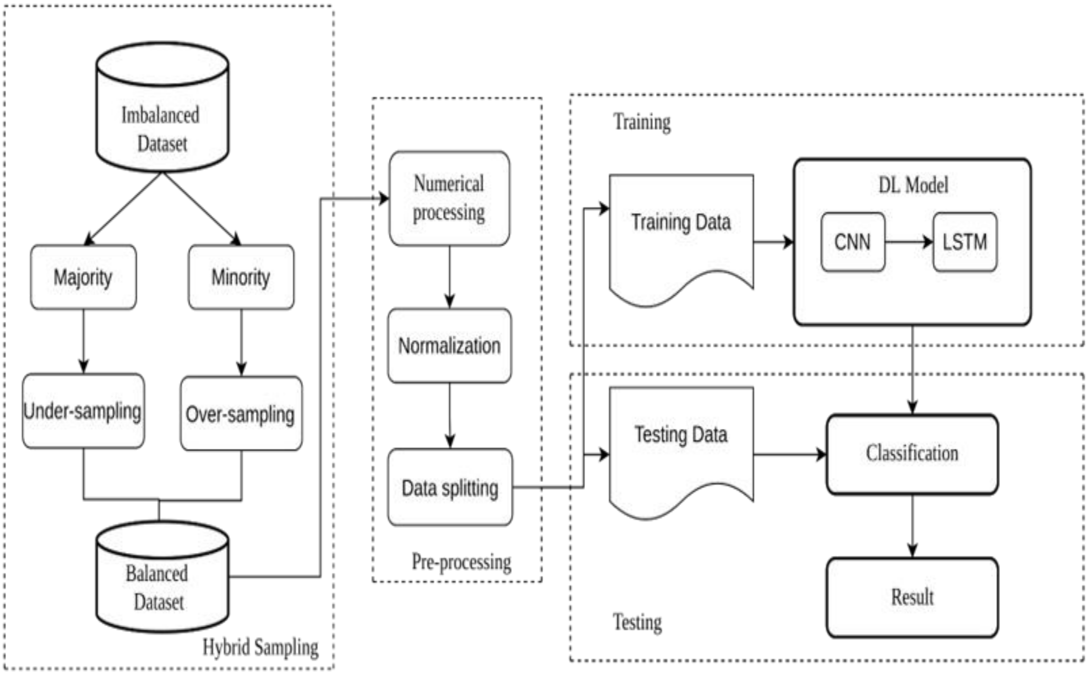


Figure 3.3 The flowchart of training the CNN-LSTM model with imbalanced data.

3.4 ANALYSIS METHODS

It is critical to assess the outcomes of a DL experiment to understand how well the model did and find any areas for improvement. This section explains how to effectively examine the results of the experiments discussed in Section 3.3.

Examining the training and validation loss curves first. These graphs depict the model's loss (or error) on the training and validation sets during the training period. If the training and validation losses decrease with time, the model is learning well. Suppose the validation loss is frequently more significant than the training loss. In that case, this might indicate overfitting, in which the model performs well on the training data but does not generalize well to the new form of data.

Second, examine the training and validation accuracies. The model learns well if the training and validation accuracies increase over time. If the validation accuracy is regularly lower than the training accuracy, this might indicate overfitting.

Third, analyze the confusion matrix. The confusion matrix is a table that displays the model's true positive, true negative, false positive, and false negative predictions. It demonstrates how well the model performs on each class in the dataset while highlighting any possible issues with its predictions.

Fourth, consider the model's predictions. Predictions made by the model on a few samples from the validation or testing set demonstrate how well the model is doing. The key to efficiently interpreting DL experiment results is carefully analyzing the model's training and validation performance and predictions on individual cases.

3.5 HYBRID DL IDS

The result of this study presents an innovative anomaly detection model designed to accurately detect deviations from normal behavior in each dataset. The model comprises several key components, each contributing to its effectiveness. Firstly, the Morris Power dataset data is transformed into images, providing a visual representation that the model can quickly analyze. Secondly, the Hurst parameter is calculated, providing critical information about the self-similarity of the data. This information is then used to train a state-of-the-art DL model, specifically a CNN and LSTM network, which can detect even subtle anomalies in the data. Finally, the model is evaluated and optimized, ensuring that it provides accurate and reliable results. Through this comprehensive approach, the model is a highly effective tool in detecting anomalies in the Morris Power dataset.

3.5.1 Dataset Transformation

The Morris Power dataset is transformed into images using the DeepInsight package (Sharma, Vans, Shigemizu, Boroevich, & Tsunoda, 2019). This powerful tool enables the conversion of non-image data into well-organized images. The process of transforming the Morris Power dataset into images is illustrated in Figure 3.4, where the numerical data is transformed into pixels, forming a visual representation of the features in the dataset. This image representation makes it easier for the ML model to analyze and identify patterns and anomalies in the data. By utilizing the DeepInsight package, this study can leverage the power of DL and achieve more accurate results in detecting anomalies in the Morris Power dataset.

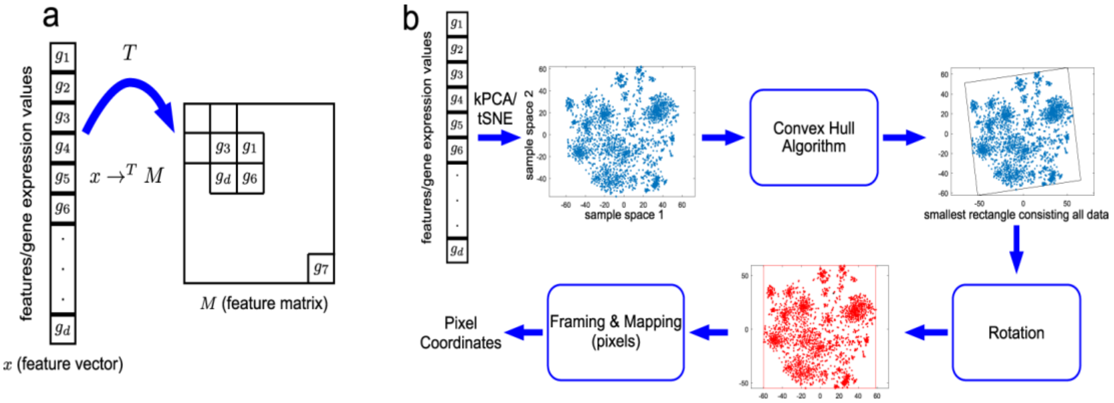


Figure 3.4 DeepInsight pipeline. (a) An illustration of transformation from feature vector to feature matrix. (b) An illustration of the DeepInsight methodology to transform a feature vector into image pixels (Sharma et al., 2019).

The DeepInsight package has been used to convert the Morris Power dataset into images, enabling the next step of the analysis process. The Hurst parameter is then calculated for each image, providing important information about the self-similarity of the data. This information is critical for the anomaly detection process, as it allows the model to identify deviations from the normal behavior represented by the Hurst parameter values calculated from the normal data. Using the Hurst parameter values as a threshold, the model can accurately detect anomalies in the Morris Power dataset, making this a vital component of the overall approach used in this study. The following section will discuss the calculation of the Hurst parameter in the next section.

3.5.2 Hurst Parameter Calculation

The Hurst parameter is a statistical measure that provides information about the self-similarity of a signal or a pattern and is commonly used in calculating fractal dimension. In the context of anomaly detection in the Morris Power dataset, the Hurst parameter is calculated for each image to determine the self-similarity of the data. The Hurst parameter can be calculated using the *R/S* analysis method, which involves dividing the signal's range by the signal's standard deviation at different time scales. The formula for the Hurst parameter can be expressed as follows:

$$H = E[log(R/S)] / log(n) \qquad (1)$$

where H is the Hurst parameter, R is the range of the data, S is the standard deviation of the signal, and n is the time scale. The calculation of the Hurst parameter involves several steps. First, the data is divided into overlapping segments of length n, and for each segment, the range (R) and standard deviation (S) are calculated. The *R/S* ratio is then calculated for each segment, and the logarithm of this ratio is taken. Finally, the expected value of the logarithm of the *R/S* ratio is calculated and divided by the logarithm of the time scale (n), resulting in the Hurst parameter.

The Hurst parameter calculation is a critical step in the anomaly detection of the Morris Power dataset. Using the Hurst parameter values obtained from the normal data as a threshold, the model can accurately distinguish between normal and abnormal behavior, providing a robust and effective approach for detecting anomalies. The Hurst parameter has proven to be a valuable tool in this study, enabling the model to detect anomalies with high accuracy and precision.

With the Hurst parameter calculation complete, the next step in the process is to build the DL model that will be used to analyze the images and detect anomalies. This study uses a CNN combined with an LSTM network to detect anomalies. The combination of these two networks provides a powerful approach for analyzing image data and detecting anomalies, leveraging the strengths of both networks to achieve improved performance. In the following section, the details of the CNN-LSTM model used in this study will be discussed in detail.

3.5.3 CNN-LSTM Model

The CNN-LSTM model architecture combines two separate networks: a CNN and an LSTM network. The CNN network helps to extract meaningful information from the input data, in this case, the network traffic data, by using multiple layers such as convolutional layers, activation functions, pooling layers, BN, dropout, and a flattened layer.

Batch Normalization (BN) is a technique that improves the training of deep learning models by addressing internal covariate shifts and stabilizing the learning process. It normalizes the activations within each layer, reducing the impact of changing input distributions during training. BN also acts as a form of regularization, mitigating the vanishing and exploding gradient problems. Overall, BN enhances training efficiency, improves model performance, and ensures a more stable and effective learning process.

The LSTM network is designed to handle sequences of data and is used to capture any temporal relationships in the data. The LSTM network comprises multiple LSTM layers and a Dropout layer, which helps prevent overfitting during training. The final prediction is made by passing the output of the LSTM network through a dense layer with softmax activation, which maps the outputs to a probability distribution over the different classes; see Table 3.1 for a summary of all layers.

In summary, the CNN-LSTM model architecture uses a combination of a CNN and an LSTM to process traffic images and predict the abnormality. The CNN component extracts relevant features from the speech signals, while the LSTM component sequentially processes the features to capture temporal dependencies.

Table 3.1 The Architecture of the CNN-LSTM

Layer	**Output Shape**
conv2d_3 (Conv2D)	(73, 73, 32)
MaxPooling 2D	(36, 36, 32)
Flatten	(41472)
Reshape	(1, 41472)
LSTM	(32)
Dense	(1)

3.6 SUMMARY

In the context of anomaly detection, this research advances the field by introducing a sophisticated approach tailored to the Morris Power dataset. Utilizing a combined CNN and LSTM architecture, the study presents a hybrid DL model designed to identify both spatial and temporal irregularities. The methodology is comprehensive. First, the Morris Power dataset is transformed into images using the DeepInsight package. This transformation allows for a visual representation, facilitating a deeper engagement with the DL model. This transformation process, from numerical data to pixels, offers an enhanced visualization of key features.

Subsequently, the research integrates the Hurst parameter calculation. This statistical measure gauges the self-similarity of the data, serving as a threshold. By comparing deviations from the norm, encapsulated by the Hurst values, the model gains a nuanced perspective to detect anomalies. At the core of the methodology is the hybrid DL model. The CNN component effectively extracts spatial features from the images, while the LSTM captures the temporal dynamics. This model is enhanced with dropout layers and batch normalization. Dropout layers prevent overfitting by ensuring the model remains versatile, while batch normalization accelerates the training process, addressing the challenges of internal covariate shift.

The contributions and novelty are evident. The integration of the CNN-LSTM model, augmented with dropout and batch normalization, is a novel endeavor for network traffic data, especially post-image transformation. Employing the DeepInsight package for anomaly detection is a unique aspect of this study. Additionally, combining the Hurst parameter with DL methodologies signifies a pioneering approach, merging statistical techniques with the power of DL. In conclusion, this thesis provides an enhanced approach to anomaly detection. By synthesizing traditional statistical methods with advanced DL techniques, the research offers valuable insights and sets a precedent for future studies in this domain.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter focuses on the results of the dataset analysis and the four experiments conducted to address the impact of dataset imbalance in developing a SCADA-based IDS. This chapter is organized as follows: Section 4.2 provides the result of the first objective: the detailed analysis of the SCADA datasets used to develop IDSs. Section 4.3 discusses the outcome of the four experiments to tackle the SCADA imbalanced datasets. Section 4.4 discusses the open issues and challenges in developing SCADA-IDSs with imbalanced datasets. Section 4.5 provides a summary of the results of this research.

4.2 SCADA DATASETS ANALYSIS

This section presents a detailed analysis of public intrusion datasets for SCADAs, focusing on how security researchers used them to develop an IDS, their results, and the effect of the dataset's drawbacks. According to (Kenyon, Deka, & Elizondo, 2020), the absence of relevant datasets is one of the major obstacles to improving IDS for industrial control systems. However, producing realistic datasets needed costly networked assets, specific traffic generators, and complicated design planning. The SCADA datasets are examined using a variety of criteria.

- How a dataset in an IDS is cleaned, trained, and evaluated.
- What are the outcomes of using the dataset to create IDSs for SCADAs?
- The drawbacks of a dataset and how it affects the development of IDSs.

Furthermore, we address the constraints and issues associated with conducting security research using the datasets examined. Before discussing the results of the analysis, it is crucial to provide a brief explanation of key concepts in SCADA datasets.

4.2.1 Critical Concepts in SCADA Datasets

Generally, SCADA datasets are divided into two categories: a) real-world and b) simulated datasets. Real-world datasets refer to data collected from actual SCADA systems in the field. These datasets may include information about the operations of the SCADA system, such as the values of process variables, the status of equipment, and the commands issued by operators. Real-world datasets can be helpful in training and evaluating SCADA IDS because they provide a realistic representation of the types of data and events that an IDS might encounter in practice. Simulated datasets, on the other hand, are created artificially and are not based on real-world data. They may be used to test or evaluate SCADA IDS in a controlled environment. Simulated datasets can be helpful because they allow researchers to test the performance of an IDS under a variety of different conditions and configurations. However, it is essential to consider the limitations of simulated datasets and how well they represent the real-world conditions that an IDS might encounter (Choi, Yun, & Kim, 2019).

The size of the dataset is regarded as a significant factor in determining the accuracy of an ML/DL model. Large datasets often improve classification performance, whereas small datasets may be overfitting (Althnian et al., 2021). Overfitting occurs when a classifier perfectly fits the training data but performs inadequately on invalidation and other data—detecting misleading patterns that will not reoccur in the future, resulting in less precise predictions.

The distribution of attacks in datasets is most likely the most challenging part of generating a SCADA dataset. If the attacks are not correctly executed, they might result in an invalid system representation or biases in the detection process (Conti, Donadel, & Turrin, 2021). The datasets used to develop an IDS should be balanced, which implies that the numbers of normal and abnormal records should be similar. Figure 4.1 depicts a balanced dataset, whereas Figure 4.2 depicts an imbalanced dataset.

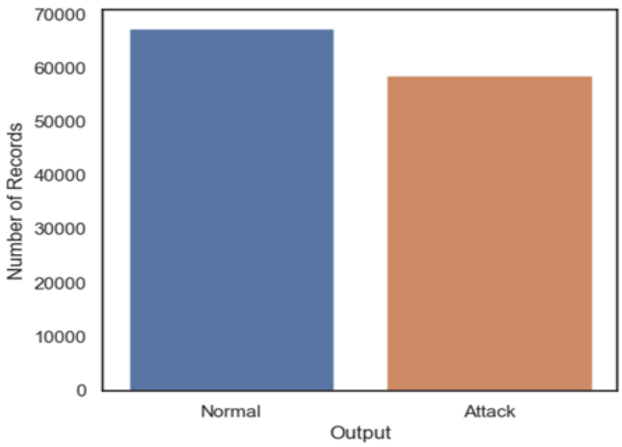


Figure 4.1 A balanced dataset from the NSL-KDD.

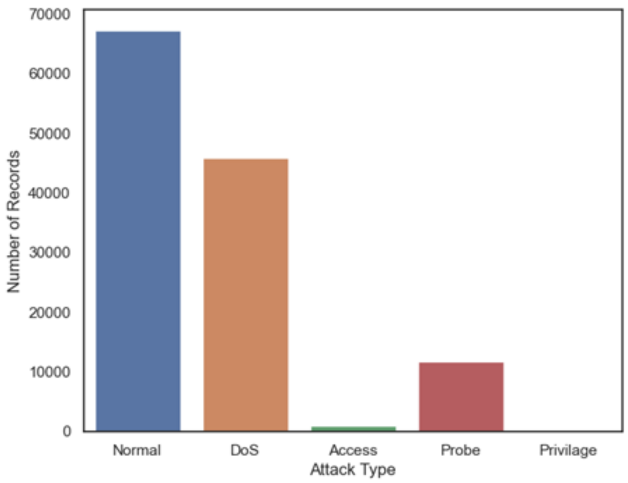


Figure 4.2 An imbalanced dataset from KDD99

It is essential to mention that the likelihood of attacks over benign records in actual networks is generally relatively low, especially when record data is collected in mass quantity over periods.

Unlike datasets generated through a testbed, attacks can be represented in various ways depending on the goals of the simulation and the capabilities of the SCADA IDS being tested. One way to represent attacks in simulated SCADA datasets is to inject anomalies or malicious events into the data. Another way to represent attacks in simulated SCADA datasets is to simulate the effects of a successful attack on the system. The characteristics of SCADA Datasets discussed in this research are described next.

4.2.2 Characteristics of SCADA Datasets

The main qualities of a SCADA dataset include the data type, whether it is actual SCADA data or simulated data generated through a testbed. The format of the data if it comes in CSV files or collected PCAP files. The protocols are one of the most critical features of the dataset, alongside the number of features, as the protocols used in a SCADA network can have significant implications for the system's security. Some standard protocols used in SCADA networks include Modbus, DNP3, TCP, UDP, HTTP, HTTPS, and ICMP. As discussed in Section 4.2.1, the size of the dataset plays a critical role in determining the performance of an ML/DL IDS model. The ANR is another critical quality in our research as it relates directly to the dataset imbalance issue.

The ANR measures the balance between normal, benign, and malicious activity in a dataset. In SCADA datasets, ANR may refer to the ratio of normal data, like the process variables and the equipment status, to data representing an attack or other malicious activity, such as abnormal values and unexpected commands. A dataset with a high ANR (a large proportion of attack data) may be more challenging for an IDS to analyze, as it may have more false positives (normal data incorrectly identified as attack data) or false negatives (attack data not detected). On the other hand, a dataset with a low ANR (a small proportion of attack data) may be less challenging for an IDS. Still, it may not provide as much information about its capabilities and limitations. Table 4.1 shows these criteria found in the datasets discussed in our research. The following sections analyze Four publicly available datasets used in developing SCADA IDSs.

Table 4.1 Characteristics of the SCADA datasets.

Dataset	**Data Type**	**Format**	**Protocols**	**Number of Features**	**ANR %**	**Size MB**
UNSW-NB15	Simulated Network Data	CSV	TCP, UDP, ICMP	49	14.5	687.2
CICIDS2017	Simulated Network Data	CSV	HTTP, HTTPS, SSH, FTP, Email	83	19.98	51,100
KDD99	Network Data	CSV	TCP, UDP, ICMP	42	67.6	743
NSL-KDD	Network Data	CSV	TCP, UDO, ICMP	41	87.06	18
ISCX2012	Simulated Network Data	PCAP, CSV	HTTP, HTTPS, FTP, SSH, and email	80	NA	78,600
Morris Gas Pipeline	Network data	CSV	Modbus	12	2.75	47
Morris Power System	Field data and device logs	CSV and ARFF	Modbus	128	245	231
Bot-IoT (5%)	Network Data	PCAP, CSV	TCP, UDP, and HTTP	29	768,980	1070
ICS cyber testbed	Simulated Network Traffic Data	CSV	S7,TCP, ARP, Telnet,	16	126	13.5

industrial OT dataset			ICMP, and HTTP			

4.2.3 Morris Gas Pipeline Dataset Analysis

This dataset is a corpus of marked RTU telematics streams generated by a gas pipeline system stored at Mississippi State University's Critical Infrastructure Protection Center in 2011 (Morris, Vaughn, & Dandass, 2011). There are 14 different files in this dataset; see Table 4.2 for an overview of the content of these files. The dataset contains records of command injection attacks, response injection attacks, and normal operation scenarios (Beaver, Borges-Hink, & Buckner, 2013).

Table 4.2 Description of files containing the Morris Gas Pipeline dataset.

File Name	**Description**
AddressScanScrubbedV2.csv	Scanning the whole network by sending packets with addresses.
FunctionCodeScanScrubbedV2.csv	A scan to find the function codes.
IllegalSetpointScrubbedV2.csv	Modifying the value of the pipeline pressure.
modbusRTU_DoSResponseInjectionV2.csv	Normal RTU communication and DoS attacks
MulticlasCommandInjectionV2.csv	Combination of normal command packets and command attacks
MulticlassResponseInjectionV2.csv	Combination of normal response packets and response attacks
PIDmodificationScrubbedV2.csv	Modifying the values of the PLCs in the control loop.
scrubbedBurstV2.csv	Sending a single value multiple times at once to the pipeline pressure.
scrubbedFastV2.csv	Sending different values successively to the pipeline pressure.
scrubbedNegativeV2.csv	Setting a negative value for the pipeline pressure is invalid.
scrubbedSetpointV2.csv	Sending incorrect value equal to the setpoint value.

scrubbedSingleV2.csv	Following an accurate response with a fraudulent one in which the gas pipeline value is manipulated to trick the PLC control loop.
scrubbedSlowV2.csv	Creating a lack of confidence in the control loop by sending different values to the pipeline pressure slowly.
scrubbedWaveV2.csv	Deceiving the control loop by sending fluctuating values to the pipeline pressure.

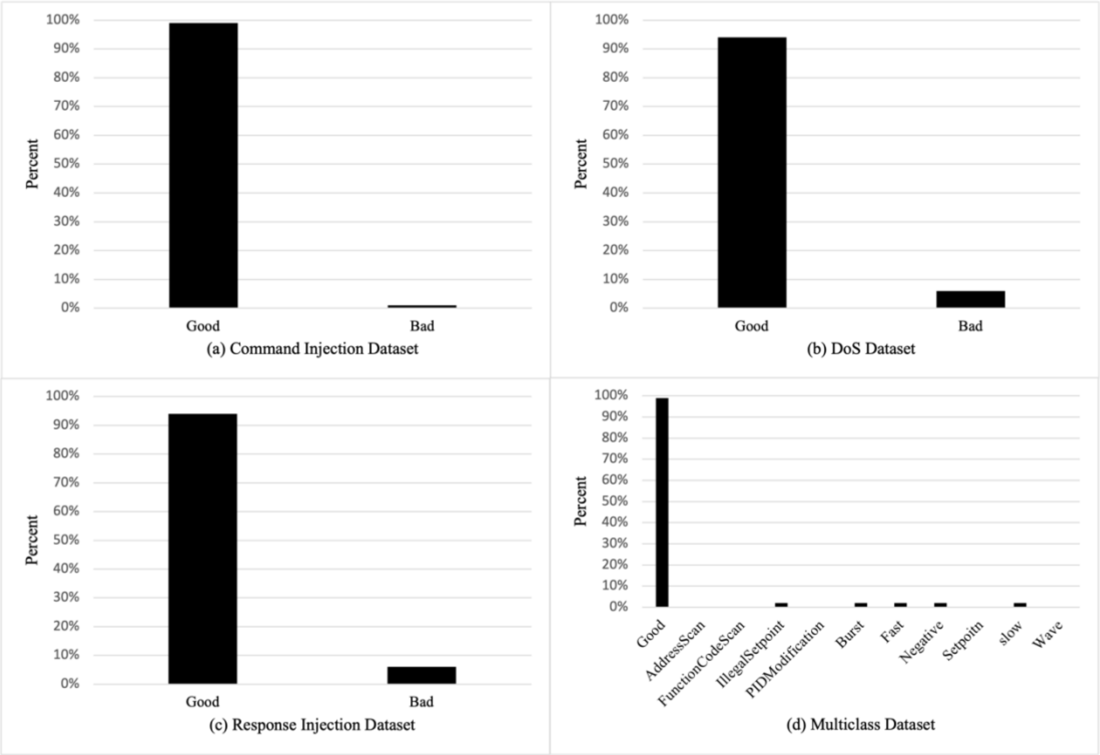


Figure 4.3 (a) Command Injection, (b) DoS Dataset, (c) Response Injection, (d) Multiclass.

The collection consists of network data in CSV format, a) Command Injection dataset, b) Denial of Service (DoS) dataset, c) Command Injection dataset, and d) Multiclass dataset. Figure 4.3 shows the class distribution in the Morris Gas Pipeline datasets; from these figures, these datasets are not balanced. The Modbus communication protocol is used in the dataset, including RTU and ASCII. The Morris Gas Pipeline dataset comprises Modbus protocol interactions between the controlling system and the HMI. The total number of normal records is 140,382, 97.32%, whereas the total number of attack records is just 3,867, 2.68% (Choi et al., 2019). These numbers indicate that this dataset is not balanced, and the detection result might be biased.

4.2.3 Morris Power System Dataset Analysis

Uttam Adhikari, Shengyi Pan, and Tommy Morris generated the Morris Power System dataset in 2014 in partnership with Borges and Justin Beaver from Oak Ridge National Laboratories' Raymond (ORNL) (C et al., 2014). It includes 37 power system event instances considering the number of Intelligent Electronic Devices (IEDs) in operation and normal/abnormal occurrences in the power grid testbed. These events are generated by different power control system devices such as IEDs, generators, and breakers. In addition to the switches and routers, which are network devices. The data is a combination of field device measurement and device logs. It consists of three datasets: a) binary dataset, b) three-class dataset, and c) multiclass dataset. The number of attack records is relatively high in the binary dataset, 55,663, around 71.02% of the total records, which is not balanced, as shown in Figure 4.4a.

Meanwhile, the normal operation records are 22,714, 29.98% (Choi et al., 2019). For the three-class dataset, the classes are also not balanced, with more than 71% of the data being attack scenarios, 23% for Natural operations, and 6% for `NoEvents`, see Figure 4.4b. In the multiclass dataset, nearly half of the records are Relay Setting Change attacks, as illustrated in Figure 4.4c.

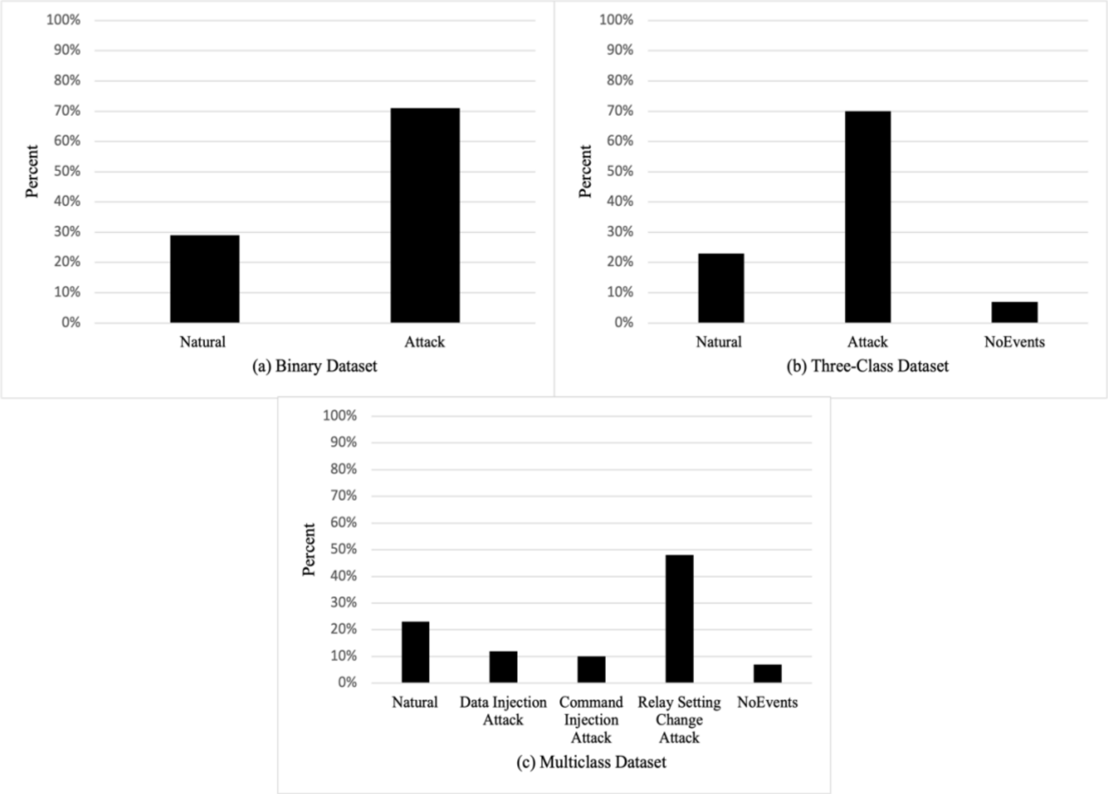


Figure 4.4 (a) Binary, (b) Three-class, (c) Multiclass.

4.2.4 Bot-IoT Dataset Analysis

Although it is an IoT dataset, analyzing and studying it for the SCADA system can be very promising. According to (Menze, 2020), 2020 Kaspersky's recent report, IoT is expected to change the state of security in industrial control systems, as per 55% of enterprises (ICS). The Bot-IoT dataset was generated in 2018 by (Koroniotis, Moustafa, Sitnikova, & Turnbull, 2018) at the Research Cyber Range lab of UNSW Canberra, with around 73 million instances in the entire dataset. Bot-IoT was built on a testbed that included different virtual computers running different operating systems, network firewalls, network taps, the Node-red tool, and the Argus network security tool. The distribution of classes in the binary and multiclass datasets is shown in Figure 4.5. The IoT devices used to generate this dataset are a) weather stations, b) smart fridges, c) Motion-activated lights, d) remotely activated garage doors, and e) smart thermostats.

It comprises several sets and subsets that vary in data format, volume, and feature quantity (Peterson, Leevy, & Khoshgoftaar, 2021). This dataset mainly has three types of features: dependent, independent, and invalid.

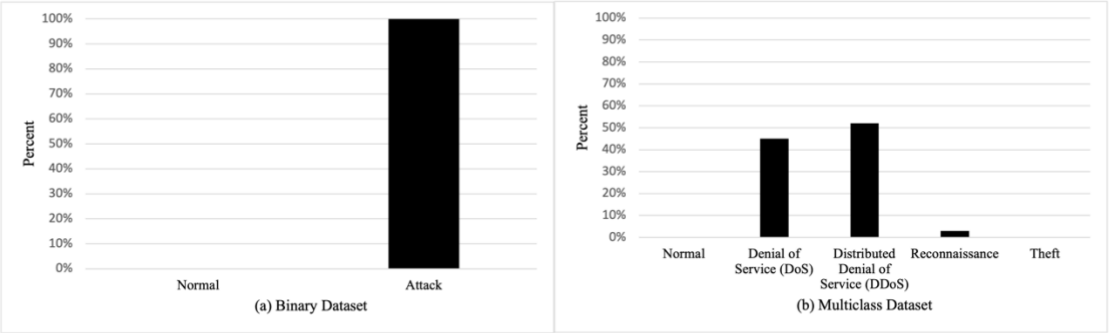


Figure 4.5 (a) Binary, (b) Three-class, (c) Multiclass

Six invalid features in the datasets include `pkSeqID, seq, time, time, saddr, and daddr.` Using invalid features will result in misleading results; for example, source addresses in the dataset are private IP addresses. If an attack has been associated with such data, then the model would classify any such private IP as malicious data.

4.2.5 CICIDS2017 Dataset Analysis

The Canadian Cyber Security Institute collected and assembled the CICIDS2017 dataset with the help of the B-Profile system at the end of 2017 (Sharafaldin et al., 2019). The dataset contains 2,830,473 network traffic samples, with benign traffic accounting for 80.30 % and attack traffic accounting for 19.7%. The categories include the most prevalent attacks, such as DoS, DDoS, Botnet, PortScan, Web Attacks, etc. The dataset collects 84 features from the generated network traffic, with the multiclass label being the last column. Furthermore, compared to publicly available datasets from 1998 to 2016, this dataset fits the 11 performance evaluation criteria. Figure 4.6 depicts the CICIDS2017 record distribution for each class.

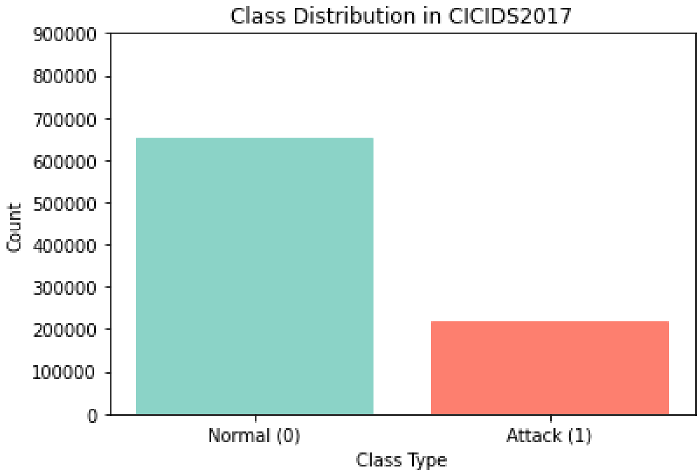


Figure 4.6 The class distribution in the CICIDS2017 dataset.

4.2.6 Issues in The Datasets

We discovered significant flaws in the generalization of attacks and anomaly event categories and frequency in several datasets examined in this study. Another issue is the lack of clarity in the labeling and descriptions of the datasets. This can be seen in the Bot-IoT and Morris Gas Pipeline datasets. The Morris GSas Pipeline dataset is scattered across 14 files, as seen in Table 4.2. Similarly, the Morris power system dataset comes in 16 CSV files, and the Bot-IoT and SCADA cyber testbed industrial OT datasets consist of 4 CSV files for each.

Another issue is the lack of description of the original data and the procedures and methodologies used to create it. As this definition implies, provenance is all about integrity, reliability, and repeatability, and it is critical for security researchers to fully understand the dataset and the environment in which it was formed (Rosa et al., 2019).

The date of event origination, the technique of event production, and the context in which the events were collected or produced are all unknown to these datasets.

All the analyzed datasets do not consider insider threats in their design. They do not consider the attacker's relation to the system in the design of such datasets. This is mainly because of data collection privacy concerns and the difficulty of simulating the full scope of hacker activities and motives. Another argument is that including insider threats and providing the public with a realistic representation of the SCADAs will give attackers a clear picture of the system and expose system weaknesses.

The lack of sustainability is another shortcoming of the datasets analyzed in this research. Datasets like the one developed by Morris in 2013 and 2014 aren't maintained or managed once deployed. The attacks and threats do not reflect reality, which is essential in a field where attack complexity is increasing regularly.

Only two out of nine datasets in Table 4.1 included PCAP files, indicating that datasets like Morris Power System, Morris Gas Pipeline, NSL-KDD, KDD99, CICIDS2017, and UNSW-NB15 do not give the original data. This is necessary for security researchers to be able to reproduce the dataset features. They include only high-level metadata, and crucial information such as timestamps, flow architecture, data flow, and protocol flags are frequently ignored. When comparing different IDS models, the absence of these features is a substantial roadblock.

An imbalanced dataset has a substantially lower number of samples from one class than the other. Almost most of these datasets include more normal data than attacks. As you can see in Section 4.2, almost all the datasets are imbalanced. In some extreme cases, the dataset is 100% attack scenarios, as in the Bot-IoT dataset; in such situations, the dataset cannot be used to train DL-based IDSs.

4.3 RESULTS OF DATASET IMBALANCE EXPERIMENTS

This section shows the results of the four experiments conducted to address the dataset imbalance problem. First is the result of training the CNN-LSTM model with the imbalanced datasets (Morris Power System and CICIDS2017). Second, is the results of training the same model. However, the datasets are balanced using the under-sampling approaches. In the Third experiment, the training data is balanced with over-sampling techniques. A combination of the two sampling methods is used in the Fourth experiment.

4.3.1 CNN_LSTM with Imbalanced Datasets

Initially, we compared dataset performance without employing any balancing strategy. The datasets were divided into two groups for binary classification: benign and attack; this is illustrated in Table 4.3. While Figure 4.7 shows the accuracy across the different number of features. The Morris Power dataset's accuracy was the highest when the features containing the same value in more than 70% of the instances were removed. On the other hand, the accuracy of the CICIDS2017 datasets remains constant with different thresholds. However, accuracy is not the best measure to evaluate performance in intrusion detection scenarios with an imbalanced dataset. Because a large portion of training data is regular traffic, the algorithms are skewed toward estimating all data as usual and disregarding the small percentage of attack events. Figure 4.8 shows the precision, recall, and F1 score.

Table 4.3 The binary classification with CNN-LSTM

Dataset	No. of Records	Type of Records	No. of Classes
Morris Power	72,073	Normal and Attack	2
CICIDS 2017	1,161,344	Normal and Attack	2

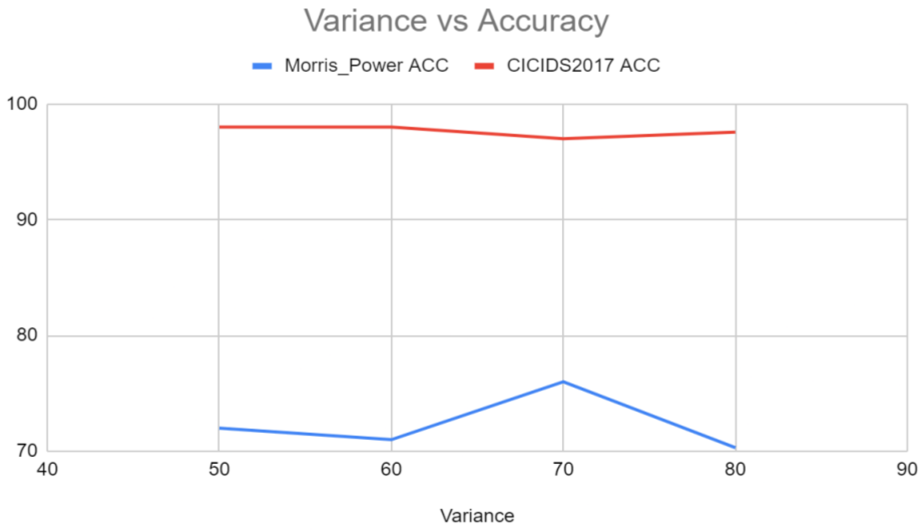


Figure 4.7 The accuracy with a different number of features

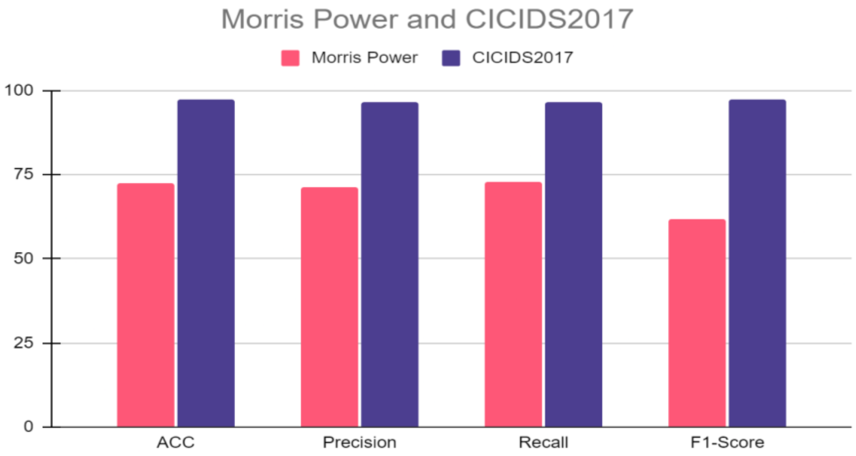


Figure 4.8 Evaluation metrics for the imbalanced datasets

The CICIDS2017 performance metrics are much better than the Morris power dataset, with approximately 97% and 75% for the F1-score. We believe this is because the CICIDS2017 has a higher ANR and better dataset quality. After comparing four values for the variance threshold, we continued our research using the threshold of 70%.

4.3.2 CNN_LSTM with Under-Sampling

The second phase of our experiment is to sample the data using under-sampling techniques, initially balancing the CIC-IDS2017 dataset with Random Under-Sampling and CNN-LSTM. The selection of features was determined by ANOVA F-value, which selected the highest score. The dataset is balanced using under-sampling algorithms such as Random Under-Sampler (RUS), One-Sided Selection, and Near Miss algorithm. The data is then divided into training and testing segments with a 70:30 split. And then, the CNN-LSTM model is trained with balanced datasets.

The results of balancing the Morris Power dataset are shown in Table 4.4. The attack class was cut by 35.3 % when a random under-sampler was used. The One-Sided Selection approach reduces the majority class by only 2.3 %. The Near Miss method produced the best results, reducing the attack class by half. This is because the Near Miss algorithm preserves the proximity of minority class instances. It selects instances from the majority class closest to those in the minority class. This approach aims to reduce class imbalance while preserving significant boundary instances. Table 4.5 shows the binary classification result using the balanced Morris Power dataset. Although the random under-sampler produced a greater F1-Score than the other algorithms, the Near Miss approach produced a higher F1-Score, which is the primary metric in our research. The performance has improved by 9 % as compared to the imbalanced dataset. The imbalanced dataset has an F1-Score of 57%, while the balanced dataset has an F1-Score of 66%.

Table 4.4 Balancing Morris Power dataset with under-sampling

Technique	Before		After	
	Normal	**Attack**	**Normal**	**Attack**
Random	15,471	38,583	15,471	25,000
One Sided Selection	15,471	38,583	15,471	37,706
Near Miss	15,471	38,583	15,471	19,338

Table 4.5 Evaluation metrics for the Morris Power dataset with under-sampling.

Technique	ACC	Precision	Recall	F1-Score
Random	71.38	51	71	59
One Sided Selection	70.91	50	71	59
Near Miss	65.89	72.07	65.67	66

Compared to the Morris Power dataset, the CICIDS2017 dataset is 18 times more prominent, and the sampling process took a long time. Tables 4.6 and 4.6 display the results of applying the under-sampling method to balance this large dataset. As far as a balanced dataset was concerned, the near-miss algorithm delivered the best results. To a maximum of 99.34 %, the model's performance is boosted by 2%. The random under-sampler achieved 96 %, while the One-Sided Selection generated an F1-Score of 97.67%. Overall, the performance of the CICIDS2017 datasets is excellent.

Table 4.6 Balancing the CICIDS2017 dataset with under-sampling

Technique	Before		After	
	Normal	Attack	Normal	Attack
Random Under Sampling	652,757	218,251	250,000	218,251
One Sided Selection	652,757	218,251	648,519	218,251
Near Miss	652,757	218,251	291,001	218,251

Table 4.7 Evaluation metrics for the CICIDS2017 dataset with under-sampling.

Technique	ACC	Precision	Recall	F1-Score
Random	96.65	94	98	96
One Sided Selection	97.34	96.62	98.04	97.67
Near Miss	99.25	99.44	99.25	99.34

4.3.3 CNN_LSTM with Over-Sampling

This section describes the third experiment, which only used over-sampling approaches to balance the datasets. Random Over-Sampler (ROS), Synthetic Minority Overs-Sampling Technique (SMOTE), and Adaptive Synthetic Sampling are the algorithms used to balance the data (ADASYN). The remaining steps are the same as detailed in section 4.3.2.

Table 4.8 displays the outcomes for the Morris Power dataset; employing ROS, the minority class is doubled in size. SMOTE oversamples the normal class, resulting in a nearly balanced dataset. ADASYN performed well when over-sampling the minority category but did not perform optimally. In terms of performance, the accuracy of all algorithms is around 71%. The difference is evident in the other metrics; for example, the SMOTE algorithm performed best in the F1-Score, scoring 64%; the detailed findings are shown in Table 4.9.

Table 4.8 Balancing Morris Power dataset with over-sampling.

Technique	**Before**		**After**	
	Normal	**Attack**	**Normal**	**Attack**
Random	15,471	38,583	30,866	38,583
SMOTE	15,471	38,583	34,724	38,583
ADASYN	15,471	38,583	32,425	38,583

Table 4.9 Evaluation metrics for the Morris Power dataset with over-sampling.

Technique	**ACC**	**Precision**	**Recall**	**F1-Score**
Random	71	51	71	59
SMOTE	70	65	70	64
ADASYN	71.37	64	71	61

In the CICIDS2017, the SMOTE algorithm outperformed ROS and ADASYN in over-sampling the minority class; Table 4.10 provides the actual values. Table 4.11 displays the outcome of the CNN-LSTMM binary classification with the oversampled dataset. The accuracy declined from 99.40 % when SMOTE was used to 93.62 % when ADASYN was used, and the F1-Score dropped from 99.46 % to 93.25 %. The drop in performance when using ADASYN oversampling in anomaly detection with a CNN model is due to several factors, including:

1. Synthetic instances from ADASYN can introduce noise, confuse the model, and hinder its ability to distinguish between normal and abnormal patterns.
2. ADASYN's effectiveness relies on diverse and representative minority class instances; otherwise, the generated synthetic instances may not accurately represent true anomalies, leading to decreased performance.
3. ADASYN can also distort decision boundaries if synthetic instances are not well-distributed or aligned with true anomalies, impacting the model's ability to detect subtle deviations.
4. There is an increased risk of overgeneralization, resulting in higher false positive rates and negatively affecting precision, recall, and F1-score metrics.

Table 4.10 Balancing CICIDS2017 dataset with over-sampling.

Technique	**Before**		**After**	
	Normal	**Attack**	**Normal**	**Attack**
Random	652,757	218,251	652,757	476,512
SMOTE	652,757	218,251	652,757	522,205
ADASYN	652,757	218,251	652,757	457,547

Table 4.11 Evaluation metrics for the CICIDS2017 dataset with over-sampling.

Technique	**ACC**	**Precision**	**Recall**	**F1-Score**
Random	99.63	99.04	99.78	99.41
SMOTE	99.47	99.43	99.49	99.46
ADASYN	93.62	92.37	99.18	93.25

4.3.4 CNN_LSTM with Hybrid-Sampling

The fourth and final experiment balanced the datasets using under-sampling and over-sampling methods. As shown in Table 4.12 for the Morris Power dataset's first coupled algorithm, SMOTE and Near-Miss. This method succeeded in balancing the Morris Power dataset. The detailed values for the evaluation metrics are provided in Table 4.13. The Morris Power dataset significantly reduced accuracy from 75% to 59%.

Table 4.12 Balancing Morris Power Dataset with Hybrid Sampling

Technique	Before		After	
	Normal	Attack	Normal	Attack
SMOTE & Near Miss	15,471	38,583	27,008	31,774
ADASYN & Near Miss	15,471	38,583	33,252	23,277

Table 4.13 Evaluation metrics for the Morris Power dataset with hybrid sampling.

Technique	ACC	Precision	Recall	F1-Score
SMOTE & Near Miss	66.69	60	67	62
ADASYN & Near Miss	69.47	56	69	59

On the other hand, the result of balancing the CICIDS 2017 with a hybrid technique is shown in Table 4.14. The performance of the binary classification model decreased. A roughly similar result is obtained when ADASYN is combined with the near-miss algorithm for hybrid balancing. The detailed values for the evaluation metrics are provided in Table 15. In the CICIDS 2017, accuracy decreased from 93.44% to 89.84%.

Table 4.14 Balancing the CICIDS2017 dataset with hybrid sampling.

Technique	Before		After	
	Normal	Attack	Normal	Attack
SMOTE & Near Miss	652,757	218,251	559,505	391,654
ADASYN & Near Miss	652,757	218,251	396,819	277,466

Table 4.15 Evaluation metrics for the CICIDS2017 dataset with hybrid sampling.

Technique	ACC	Precision	Recall	F1-Score
SMOTE & Near Miss	93.44	94.05	93	93.32
ADASYN & Near Miss	89.84	90.15	89.36	89.58

4.4 RESULTS OF ANOMALY DETECTION WITH SELF-SIMILARITY

The results of the transformation of the Morris Power dataset into image representations and the subsequent analysis utilizing DeepInsight are presented in this section. A sample of normal traffic packet data has been transformed and depicted in Figure 4.9, while Figure 4.10 displays a sample of attack data. The visual comparison of these images reveals that there is an increase in the pixel intensity in the top right corner of the attack data image (Figure 4.10).

These image representations of the Morris Power dataset were also utilized to detect anomalies and classify the data into normal or attack classes. The Hurst parameter calculation was also performed on the dataset to further aid anomaly detection. The results of the Hurst parameter calculation are discussed next.

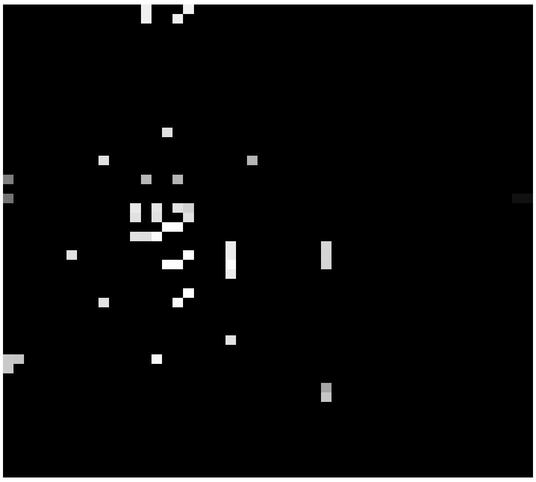


Figure 4.9 Example of the normal packet in the Morris Power dataset converted into an image.

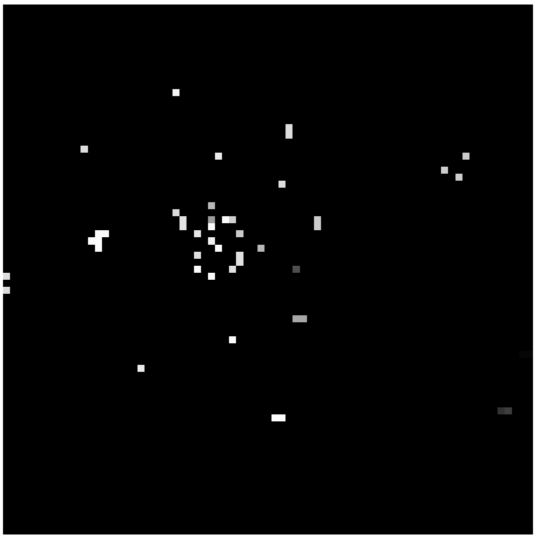


Figure 4.10 Example of an attack packet in the Morris Power dataset converted into an image.

The dataset underwent a thorough conversion and was divided into three directories: Natural, Attack, and Test. To evaluate the statistical self-similarity of the images, the Hurst parameter has been calculated for each image within each of these directories.

The Hurst parameter for images in the Natural folder is a benchmark for determining the "abnormal" behavior threshold. The Hurst parameter of the images in the Attack and Test folders has been determined accordingly, and the images within the Test folder have been classified as "abnormal" if their Hurst parameter falls below the established threshold.

The analysis of the Hurst parameters for the Attack and Natural data has been visualized in Figure 4.11, providing an in-depth look at the Mean, Min, and Max values. Figure 4.12 compares the Hurst parameters of the Natural and Attack data, comprehensively evaluating the differences between the two datasets. Finally, Figure 4.13 presents a histogram comparison between the Natural and Attack data against all data, offering a graphical representation of the distribution of Hurst parameters for each category.

This systematic and data-driven approach allows for a robust evaluation of the Hurst parameter, providing a quantitative measurement of the statistical self-similarity of the images and facilitating the classification of "abnormal" behavior.

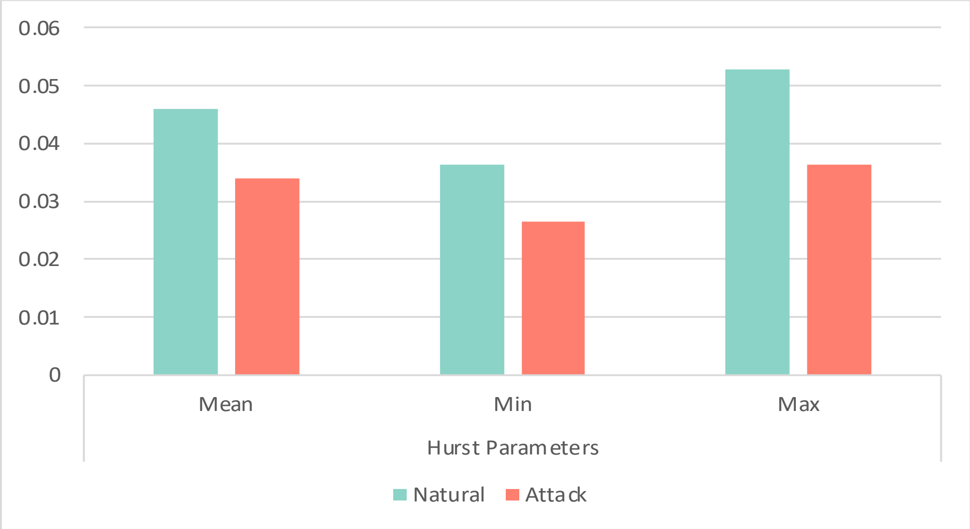


Figure 4.11 Statistical measures for the Hurst values calculated for the Morris Power Dataset.

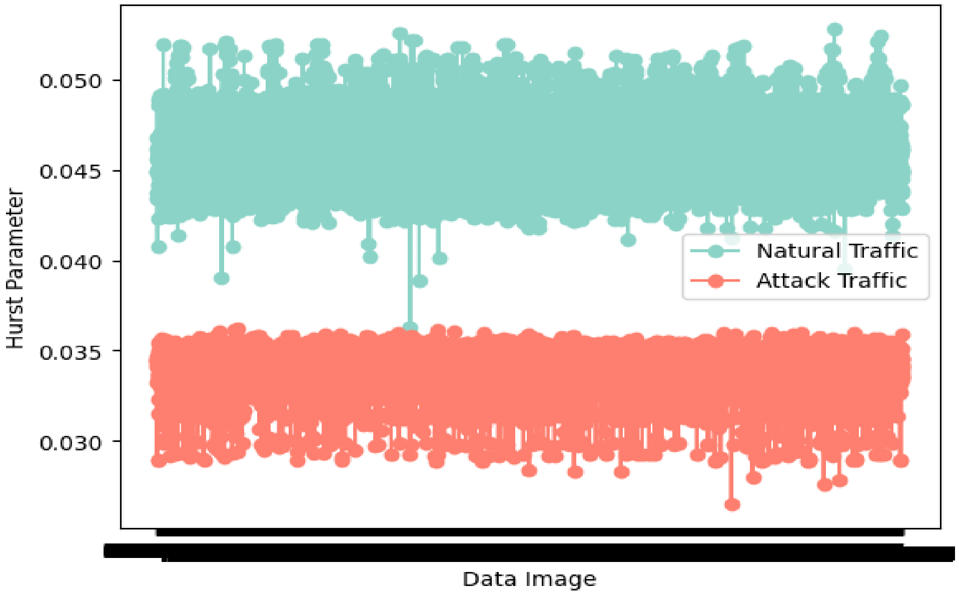


Figure 4.12 Comparison of Hurst parameters for Natural vs. Attack traffic images in Morris Power Dataset

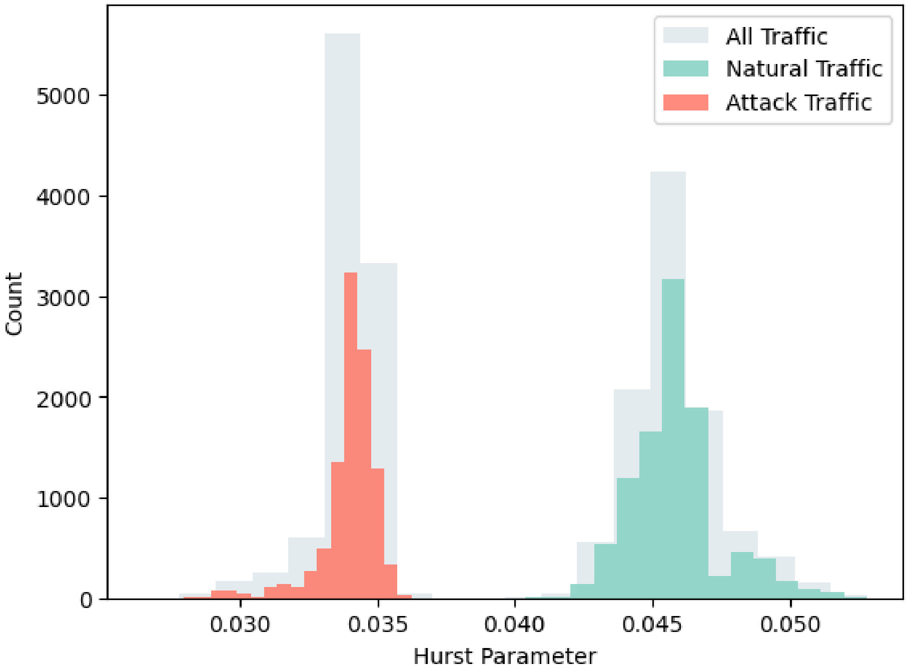


Figure 4.13 Comparison of Hurst parameters for Natural vs. Attack traffic images in Morris Power Dataset.

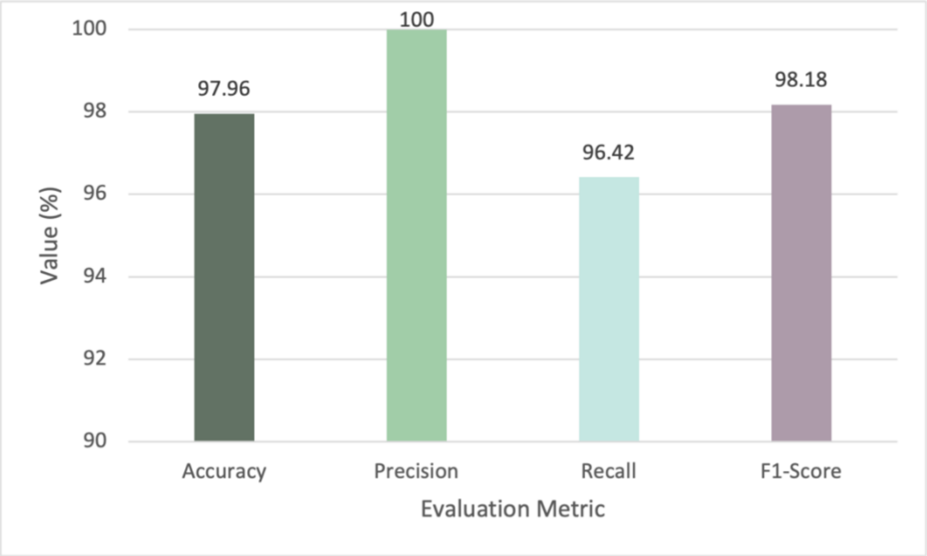


Figure 4.14 Evaluation Metrics of Hurst IDS using Morris Power Dataset.

The anomaly detection results using the CNN-LSTM model are presented in Figure 4.15. This figure comprehensively evaluates the model's performance, including accuracy, precision, recall, and F1-score, which are widely used metrics in ML and computer vision. From the bar chart, the CNN-LSTM model has achieved high performance in detecting network traffic anomalies.

A combination of the Hurst detector and the CNN-LSTM model is employed to enhance the performance further. BN and dropout layers are used to enhance the CNN-LSTM model. The Hurst detector is used to identify any abnormal network traffic data, and then the detected data is passed to the CNN-LSTM model for double confirmation. This combined approach improves the overall accuracy and robustness of the system, making it more effective in detecting network anomalies.

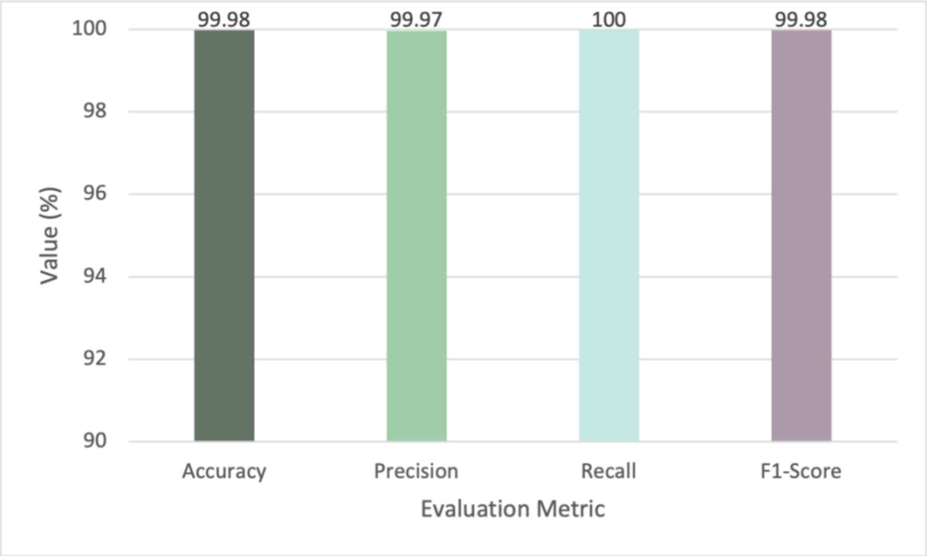


Figure 4.15 Evaluation Metrics of CNN-LSTM IDS using Morris Power Dataset.

The hybrid model that combines self-similarity with the Hurst parameter and the CNN-LSTM model achieved outstanding results in detecting anomalies in network traffic data. As shown in Figure 4.16, the model achieved perfect accuracy, precision, recall, and F1-score. It effectively identifies abnormal traffic patterns with high accuracy and low false positive and false negative rates. This is a testament to the model's ability to effectively integrate the Hurst parameter's ability to detect long-range correlations in network traffic data and the CNN-LSTM's capacity to learn spatial and temporal dependencies from image-based network traffic data. These results demonstrate the effectiveness of this hybrid approach in detecting anomalies in network traffic and highlight the importance of considering both statistical and neural network-based approaches in network traffic analysis.

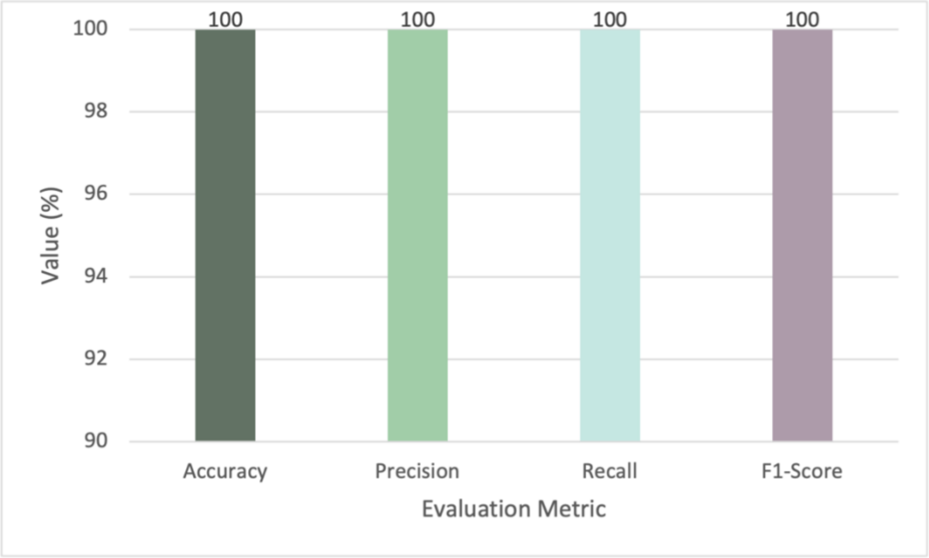


Figure 4.16 Evaluation Metrics of Hybrid Self-similarity and CNN-LSTM IDS using Morris Power Dataset.

4.5 OPEN ISSUES AND CHALLENGES

The dataset is the backbone of the development of an IDS. Therefore, a well-designed IDS model requires a well-designed dataset. The dataset is usually generated through a testbed or a real-time collection of a running SCADA system. In the case of a testbed, the dataset should fully describe the architecture and the behavior of the testbed. Since most current IDS models are based on supervised learning techniques, the labeling procedure must be fully described in the dataset's documentation for researchers to act accordingly. Moreover, the label should include the attack type and phase, which is crucial for the recovery process after an attack. A system may consider this part of the attack if it is not accurately labeled as a recovery.

Furthermore, to allow for simple and efficient usage of the dataset, the dataset should include a clear and comprehensive description of the testbed architectural design and a thorough description of the dataset characteristics. One of the critical metrics for evaluating a dataset should be its documentation. System settings, assumptions, and an extensive explanation of the attacks included in the testbed are just a few of these features.

Another challenge of developing an IDS with imbalanced datasets is that there is no systematic method of selecting which DL model to use because each has its own set of characteristics that make it ideal for application. All the papers produced an acceptable deep-learning method. Although each paper is evaluated with different metrics like accuracy, recall, and F1 score, the classification model's evaluation process is not standard. This makes it difficult to compare two approaches to the same problem. A straightforward procedure for the assessment of a DL classification algorithm is required.

While the results achieved by the hybrid model are impressive, it is essential to note that overfitting remains a possibility. Overfitting occurs when a model performs well on the training data but poorly on unseen data.

4.6 SUMMARY

IDS are essential tools for protecting critical infrastructures, such as the industrial control systems used in SCADA networks. However, developing an IDS for a SCADA system can be challenging, mainly when dealing with imbalanced datasets. Several approaches can be taken to address the issue of imbalanced datasets in developing an IDS for a SCADA system. One approach is to use under-sampling or oversampling techniques to balance the dataset. Under-sampling involves reducing the number of instances in the majority class, while over-sampling involves generating synthetic instances of the minority class.

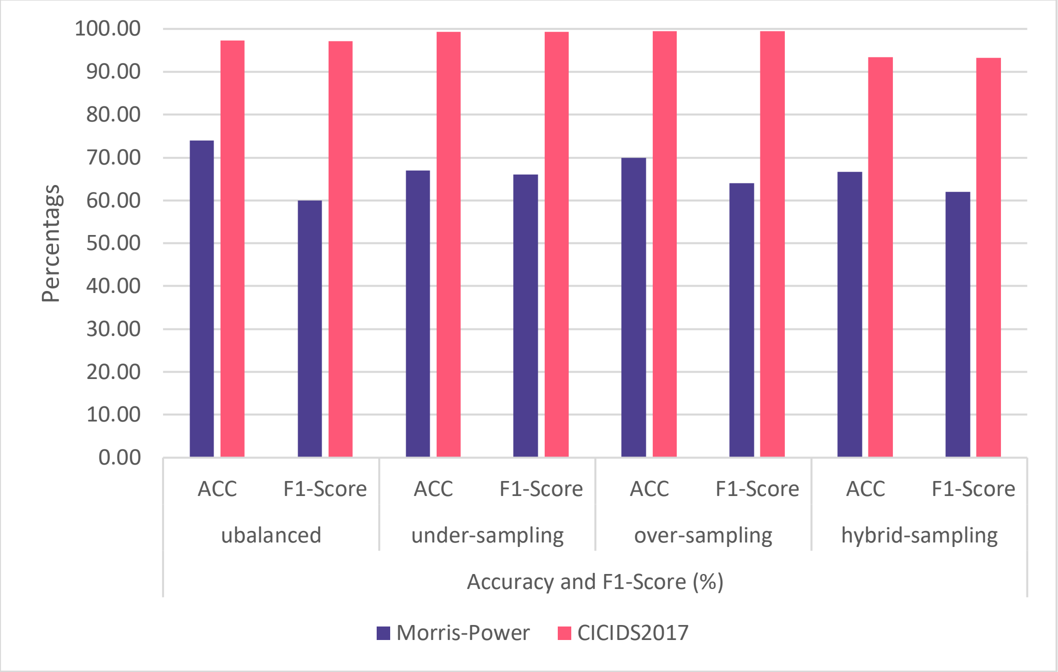


Figure 4.17 The F1-score and accuracy of both datasets in all four experiments.

The overall result is shown in Figure 4.9, and it is clear from the pattern that when using the Morris Power dataset, the CNN-LSTM model performs quite modestly. This is due to the dataset's small size, only around 72,000. The outcome may differ if an ML algorithm is used. DL techniques, however, need a more extensive dataset. The results using the imbalanced CICIDS2017 dataset were satisfactory. When over-sampling is used exclusively, the best outcome is obtained. Figure 4.9 demonstrates that the hybrid-sampling method did not produce reliable outcomes. This is because the dataset is distorted by removing records from the majority class and adding fake data to the minority class.

This research enhanced the CNN algorithm by balancing the Morris Power System and CICIDS2017 datasets. The results shown in Figure 4.9 indicated a slight improvement in the F1-Score for both the Morris Power dataset and the CICIDS2017 dataset.

CHAPTER FIVE

CONCLUSION

This section provides the conclusions of this research and the recommendations for future research. Because it is directly tied to real life, SCADA security is a contentious issue in ongoing security research. On the topic of developing resilient and reliable IDSs, this work offers a thorough investigation of the publicly available SCADA datasets. The research provides conclusions and recommendations on SCADA security and developing reliable and resilient intrusion detection systems. The results show that the security of SCADA systems requires a well-designed measure, and the integration of advanced technology and algorithms can improve their security. The study presents a novel combination of the Hurst Detector and the CNN-LSTM model to detect network anomalies, which showed exceptional performance and demonstrated the effectiveness of multi-model analysis in network security. Additionally, a comprehensive security strategy is essential to deploying intrusion detection sensors, signature-based and anomaly-based detection methods, secure communication protocols, and robust authentication and authorization processes. With continued research and development, the integration of cutting-edge technologies and a comprehensive security strategy will ensure the future of SCADA systems is more secure and resilient.

The future of SCADA security is bright with the advancement of technology and artificial intelligence. The results of this study on network traffic anomaly detection have demonstrated the potential for improving SCADA security through advanced algorithms and innovative approaches. With continued research and development, we can expect the integration of cutting-edge technologies like Zero-Knowledge Proof

(ZKP), Privacy by Design (PbD), blockchain, graph-based techniques, and stable diffusion models to enhance the security of SCADA systems further.

By incorporating additional network traffic features and exploring alternative neural network architectures, such as CNNs and RNNs, the performance of anomaly detection models will continue to improve. Integrating unsupervised learning techniques like autoencoders and clustering algorithms will also play a crucial role in enhancing the overall performance of these models.

In addition, a comprehensive security strategy that includes deploying IDS sensors, using signature-based and anomaly-based detection methods, secure communication protocols, and robust authentication and authorization processes will be essential in keeping SCADA systems secure. Network segmentation and access control will also help limit the spread of cyber-attacks and make it more challenging for attackers to access sensitive system areas.

As cyber threats continue to evolve, staying updated with the latest information and trends is crucial to defend against them effectively. By combining cutting-edge technology with a comprehensive security strategy, we can look forward to a future where SCADA systems are more secure and resilient than ever.

REFERENCES

Abid, A., & Jemili, F. (2020). Intrusion Detection based on Graph oriented Big Data Analytics. *Procedia Computer Science*, *176*, 572–581. Elsevier B.V.

Abokifa, A. A., Haddad, K., Lo, C., & Biswas, P. (2019). Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems via Machine Learning–Based Anomaly Detection Techniques. *Journal of Water Resources Planning and Management*, *145*(1), 04018089.

Aleesa, A. M., Younis, M., Mohammed, A. A., & Sahar, N. M. (2021). Deep-Intrusion Detection System With Enhanced Unsw-Nb15 Dataset Based On Deep Learning Techniques. In *Journal of Engineering Science and Technology* (Vol. 16).

Alladi, T., Chamola, V., Rodrigues, J. J. P. C., & Kozlov, S. A. (2019, November 2). Blockchain in smart grids: A review on different use cases. *Sensors (Switzerland)*, Vol. 19. MDPI AG.

Alladi, T., Chamola, V., & Zeadally, S. (2020, April 1). Industrial Control Systems: Cyberattack trends and countermeasures. *Computer Communications*, Vol. 155, pp. 1–8. Elsevier B.V.

Althnian, A., alsaeed, D., Al-Baity, H., Samha, A., Dris, A. Bin, Alzakari, N., … Kurdi, H. (2021). Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Applied Sciences (Switzerland)*, *11*(2), 1–18.

Ani, U. P. D., He, H. (Mary), & Tiwari, A. (2017). Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective. *Journal of Cyber Security Technology*, *1*(1), 32–74.

Ateş, Ç., Özdel, S., & Anarım, E. (2020). Graph–Based anomaly detection using fuzzy clustering. *Advances in Intelligent Systems and Computing*, *1029*, 338–345. Springer Verlag.

Aziz, M. N., & Ahmad, T. (2021). CLUSTERING UNDER-SAMPLING DATA FOR IMPROVING THE PERFORMANCE OF INTRUSION DETECTION SYSTEM. In *Journal of Engineering Science and Technology* (Vol. 16).

Beaver, J. M., Borges-Hink, R. C., & Buckner, M. A. (2013). An evaluation of machine learning methods to detect malicious SCADA communications. *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, *2*, 54–59. IEEE Computer Society.

C, R., Hink, B., Beaver, J. M., Buckner, M. A., Morris, T., Adhikari, U., & Pan, S. (2014). *Machine Learning for Power System Disturbance and Cyber-attack Discrimination*.

Cavoukian, A., Polonetsky, J., & Wolf, C. (2010). Smartprivacy for the Smart Grid: embedding privacy into the design of electricity conservation. *Identity in the Information Society*, *3*(2), 275–294.

Chawla, A., Lee, B., Fallon, S., & Jacob, P. (2019). Host Based Intrusion Detection System with Combined CNN/RNN Model. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11329 LNAI*, 149–158. Springer Verlag.

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).

Choi, S., Yun, J. H., & Kim, S. K. (2019). A comparison of ICS datasets for security research based on attack paths. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11260 LNCS*, 154–166. Springer Verlag.

Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in iot. *Procedia Computer Science, 167*, 1561–1573. Elsevier B.V.

Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., … Bian, L. (2017). Botnet detection using graph-based feature clustering. *Journal of Big Data*, *4*(1).

Coffey, K., Maglaras, L. A., Smith, R., Janicke, H., Ferrag, M. A., Derhab, A., … Yousaf, A. (2018). *Vulnerability Assessment of Cyber Security for SCADA Systems*.

Conti, M., Donadel, D., & Turrin, F. (2021). *A Survey on Industrial Control System Testbeds and Datasets for Security Research*.

Deep Learning With Tensorflow 2.0, Keras and Python | Codebasics. (2021). Retrieved December 18, 2022.

Derhab, A., Guerroumi, M., Gumaei, A., Maglaras, L., Amine Ferrag, M., Mukherjee, M., & Khan, F. A. (2019). Blockchain and Random Subspace Learning-Based IDS for SDN-Enabled Industrial iot Security. *Sensors (Switzerland)*, *19*(14).

Farwell, J. P., & Rohozinski, R. (2011). Stuxnet and the future of cyber war. *Survival, 53*(1), 23–40.

Goh, J., Adepu, S., Junejo, K. N., & Mathur, A. (2017). A dataset to support research in the design of secure water treatment systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10242 LNCS*, 88–99. Springer Verlag.

Gu, J., & Lu, S. (2021). An effective intrusion detection approach using SVM with naïve Bayes feature embedding. *Computers and Security*, *103*.

Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2020). A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Sciences*, *513*, 386–396.

Husaini, M. A. S. Al, Habaebi, M. H., Hameed, S. A., Islam, Md. R., & Gunawan, T. S. (2020). A Systematic Review of Breast Cancer Detection Using Thermography and Neural Networks. *IEEE Access*, *8*, 208922–208937.

Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., & Sun, J. (2017). *Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning*.

Jamil, S., Ur Rahman, M., & Fawad. (2022). A Comprehensive Survey of Digital Twins and Federated Learning for Industrial Internet of Things (iiot), Internet of Learning for Industrial Internet of Things (iiot), Internet of Vehicles (iov) and Internetofof Drones (iod). *Applied System Innovation*, *5*(3).

KDD Cup 1999 Data. (2021). Retrieved March 31, 2021, from

Kenyon, A., Deka, L., & Elizondo, D. (2020, December 1). Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. *Computers and Security*, Vol. 99. Elsevier Ltd.

Khan, R. U., Zhang, X., Alazab, M., & Kumar, R. (2019). An improved convolutional neural network model for intrusion detection in networks. *Proceedings - 2019 Cybersecurity and Cyberforensics Conference, CCC 2019*, 74–77. Institute of Electrical and Electronics Engineers Inc.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics (Switzerland)*, *8*(11).

Kim, H. (2012). Security and vulnerability of SCADA systems over ip-based wireless sensor networks. *International Journal of Distributed Sensor Networks*, Vol. 2012.

Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016). *Long Short Term Memory Recurrent neuralnetwork Classifier for Intrusion Detection*.

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2018). *Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-iot Dataset*.

Kwon, S., Yoo, H., & Shon, T. (2020). IEEE 1815.1-Based power system security with bidirectional RNN-Based network anomalous attack detection for cyber-physical system. *IEEE Access*, *8*, 77572–77586.

Lee, R. M., Assante, M. J., & Conway, T. (2014). *ICS-CPPE-case-Study-2-German-Steelworks_Facility*.

Li, Y., Ma, R., & Jiao, R. (2015). A hybrid malicious code detection method based on deep learning. *International Journal of Security and Its Applications*, *9*(5), 205–216.

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., Mcclung, D., … Zissman, M. A. (1999). *Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation**.

Mahdavifar, S., & Ghorbani, A. A. (2019). Application of deep learning to cybersecurity: A survey. *Neurocomputing*, *347*, 149–176.

Mahmoud, A., El-Kilany, A., Ali, F., & Mazen, S. (2021). TGT: A Novel Adversarial Guided Oversampling Technique for Handling Imbalanced Datasets. *Egyptian Informatics Journal*, *22*(4), 433–438.

Marir, N., Wang, H., Feng, G., Li, B., & Jia, M. (2018). Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark. *IEEE Access*, *6*, 59657–59671.

Menze, T. (2020). *The State Of Industrial Cybersecurity In The Era Of Digitalization*. Retrieved from

Miah, O., Khan, S., Shatabda, S., & Md.Farid Dewan. (2019). *Improving Detection Accuracy for Imbalanced Network Intrusion Classification using Cluster-based Under-sampling with Random Forests*.

Mishra, N. K., & Singh, P. K. (2021). Feature construction and smote-based imbalance handling for multi-label learning. *Information Sciences*, *563*, 342–357.

Mohamed, A.-R., Dahl, G. E., & Hinton, G. (2010). *Acoustic Modeling using Deep Belief Networks*.

Morris, T., Vaughn, R., & Dandass, Y. S. (2011). A testbed for SCADA control system cybersecurity research and pedagogy. *ACM International Conference Proceeding Series*.

Moustafa, N. (2021). The UNSW-NB15 dataset. Retrieved March 31, 2021, from

Mubarak, S., Habaebi, M. H., Islam, M. R., Balla, A., Tahir, M., Elsheikh, E. A. A., & Suliman, F. M. (2022). Industrial datasets with ICS testbed and attack detection using machine learning techniques. *Intelligent Automation and Soft Computing*, *31*(3).

Mulay, S. A., Devale, P. R., & Garje, G. V. (2010). Intrusion Detection System Using Support Vector Machine and Decision Tree. *International Journal of Computer Applications*, *3*(3), 40–43.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1).

Nasser, M., Ahmad, R., Yassin, W., Hassan, A., Zainal, Z., Salih, N., & Hameed, K. (2018). Cyber-Security Incidents: A Review Cases in Cyber-Physical Systems. *International Journal of Advanced Computer Science and Applications*, *9*(1), 499–508.

O'Connor, Y., Rowan, W., Lynch, L., & Heavin, C. (2017). Privacy by Design: Informed Consent and Internet of Things for Smart Health. *Procedia Computer Science*, *113*, 653–658. Elsevier B.V.

Onan, Aytuğ. (2019). Consensus Clustering-Based Undersampling Approach to Imbalanced Learning. *Scientific Programming*, *2019*.

Onan, Aytug, & korukoglu, S. (2017). A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*, *43*(1), 25–38.

Pan, S., Morris, T., & Adhikari, U. (2015). Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data. *IEEE Transactions on Industrial Informatics*, *11*(3), 650–662.

Pedraza, J., Patricio, M. A., de Asís, A., & Molina, J. M. (2013). Privacy-by-design rules in face recognition system. *Neurocomputing*, *109*, 49–55.

Peterson, J. M., Leevy, J. L., & Khoshgoftaar, T. M. (2021). A Review and Analysis of the Bot-iot Dataset. *Proceedings - 15th IEEE International Conference on Service-Oriented System Engineering, SOSE 2021*, 20–27. Institute of Electrical and Electronics Engineers Inc.

Pliatsios, D., Sarigiannidis, P., Lagkas, T., & Sarigiannidis, A. G. (2020). A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics. *IEEE Communications Surveys and Tutorials*, *22*(3), 1942–1976.

Pourhabibi, T., Ong, K. L., Kam, B. H., & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, *133*.

Rosa, L., Freitas, M., Mazo, S., Monteiro, E., Cruz, T., & Simoes, P. (2019). A Comprehensive Security Analysis of a SCADA Protocol: From OSINT to Mitigation. *IEEE Access*, *7*, 42156–42168.

Saxe, J., & Berlin, K. (2017). *Expose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious urls, File Paths and Registry Keys*. Retrieved from

Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2019). A Detailed Analysis of the CICIDS2017 Data Set. *Communications in Computer and Information Science*, *977*, 172–188. Springer Verlag.

Sharma, A., Vans, E., Shigemizu, D., Boroevich, K. A., & Tsunoda, T. (2019). Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific Reports*, *9*(1).

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers and Security*, *31*(3), 357–374.

Staudemeyer, R. (2015). *Applying long short-term memory recurrent neural networks to intrusion detection*. 136–154.

Suaboot, J., Fahad, A., Tari, Z., Grundy, J., Mahmood, A. N., Almalawi, A., … Drira, K. (2020). A Taxonomy of Supervised Learning for idss in SCADA Environments. *ACM Computing Surveys*, *53*(2).

Taormina, R., Galelli, S., Tippenhauer, N. O., Salomons, E., Ostfeld, A., Eliades, D. G., … Ohar, Z. (2018). Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. *Journal of Water Resources Planning and Management*, *144*(8), 04018048.

Tavallaee, M., Bagheri, E., Lu, W., & A. Ghorbani, A. (2009). *A Detailed Analysis of the KDD CUP 99 Data Set*. IEEE.

The Bot-iot Dataset | UNSW Research. (n.d.). Retrieved December 22, 2021, from

The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. — Stratosphere IPS. (n.d.). Retrieved March 27, 2021

Tian, Q., Han, D., Li, K. C., Liu, X., Duan, L., & Castiglione, A. (2020). An intrusion detection approach based on improved deep belief network. *Applied Intelligence*, *50*(10), 3162–3178.

Tommy Morris - Industrial Control System (ICS) Cyber Attack Datasets. (n.d.). Retrieved December 22, 2021.

Tsai, C. F., Lin, W. C., Hu, Y. H., & Yao, G. T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences*, *477*, 47–54.

Vargas, H., Lozano-Garzon, C., Montoya, G. A., & Donoso, Y. (2021). Detection of security attacks in industrial iot networks: A blockchain and machine learning approach. *Electronics (Switzerland)*, *10*(21).

Wang, C., Wang, B., Liu, H., & Qu, H. (2020). Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network. *Wireless Communications and Mobile Computing*, *2020*.

Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2017). HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection. *IEEE Access*, *6*, 1792–1806.

Wang, Z., Xie, W., Wang, B., Tao, J., & Wang, E. (2021, May 1). A survey on recent advanced research of cps security. *Applied Sciences (Switzerland)*, Vol. 11. MDPI AG.

Wu, C., Liu, Y., Wu, F., Liu, F., Lu, H., Fan, W., & Tang, B. (2020). A hybrid intrusion detection system for iot applications with constrained resources. *International Journal of Digital Crime and Forensics*, *12*(1), 109–130.

Wu, K., Chen, Z., & Li, W. (2018). A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. *IEEE Access*, *6*, 50850–50859.

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., … Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access*, *6*, 35365–35381.

Xu, C., Shen, J., Du, X., & Zhang, F. (2018). An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units. *IEEE Access*, *6*, 48697–48707.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, *5*, 21954–21961.

Yu, S. J., Koh, P., Kwon, H., Kim, D. S., & Kim, H. K. (2016). Hurst Parameter based Anomaly Detection for Intrusion Detection System. *2016 IEEE International Conference on Computer and Information Technology*, 788.

Yu, Y., Long, J., & Cai, Z. (2017). Network Intrusion Detection through Stacking Dilated Convolutional Autoencoders. *Security and Communication Networks*, *2017*.

Zavrak, S., & Iskefiyeli, M. (2020). Anomaly-Based Intrusion Detection from Network Flow Features Using Variational Autoencoder. *IEEE Access*, *8*, 108346–108358.

Zhang, G., Wang, X., Li, R., Song, Y., He, J., & Lai, J. (2020). Network Intrusion Detection Based on Conditional Wasserstein Generative Adversarial Network and Cost-Sensitive Stacked Autoencoder. *IEEE Access*, *8*, 190431–190447.

Zhang, H., Huang, L., Wu, C. Q., & Li, Z. (2020). An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Computer Networks*, *177*.

Zolfi, H., Ghorbani, H., & Ahmadzadegan, M. H. (2019). Investigation and classification of cyber-crimes through IDS and SVM algorithm. *Proceedings of the 3rd International Conference on I-SMAC iot in Social, Mobile, Analytics and Cloud, I-SMAC 2019*, 180–187. Institute of Electrical and Electronics Engineers

APPENDIX

LIST OF PUBLICATIONS

1. Balla, A., Habaebi, M. H., Elsheikh, E. A. A., Islam, Md. R., & Suliman, F. M. (2023). The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems. Sensors, 23(2), 758. doi.org/10.3390/s23020758. Q1 IF=3.847

2. Balla, A., Habaebi, M. H., Islam, M. R., Mubarak, S. (2022). Applications of deep learning algorithms for supervisory control and Data Acquisition Intrusion Detection System. Cleaner Engineering and Technology, 9, 100532. doi:10.1016/j.clet.2022.100532. CiteScore=0.9

3. Babiker, A. B., Habaebi, M. H., Mubarak, S., & Islam, Md. R. (2023). A detailed analysis of Public Industrial Control System datasets. International Journal of Security and Networks, 18(4), 245–263. doi:10.1504/ijsn.2023.135511.

4. A. Balla, M. H. Habaebi, E. A. A. Elsheikh, M. R. Islam, F. E. M. Suliman and S. Mubarak, "Enhanced CNN-LSTM Deep Learning for SCADA IDS Featuring Hurst Parameter Self-Similarity," in IEEE Access, vol. 12, pp. 6100-6116, 2024, doi: 10.1109/ACCESS.2024.3350978.

5. S. Mubarak, M. Hadi Habaebi, M. Rafiqul Islam, A. Balla, M. Tahir et al., "Industrial datasets with ics testbed and attack detection using machine learning techniques," Intelligent Automation & Soft Computing, vol. 31, no.3, pp. 1345–1360, 2022. Q3 IF= 3.401

6. Sinil Mubarak, Mohamed Hadi Habaebi, Asaad Balla, Md Rafiqul Islam, Elfatih A. A. Elsheikh, F. M. Suliman, “ICS SCADA Cyber-attacks Detection and Forecast with Deep Learning Algorithms” Elsevier IoT Journal Q1 IF=5.711 (under review)