5-12-2023

Anomaly detection in SCADA systems using machine learning

Eric Kudjoe Fiah
Mississippi State University, ef588@msstate.edu

Follow this and additional works at: https://scholarsjunction.msstate.edu/td

Recommended Citation
https://scholarsjunction.msstate.edu/td/5824

This Graduate Thesis - Open Access is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact scholcomm@msstate.libanswers.com.
Anomaly detection in SCADA systems using machine learning

By

Eric Kudjoe Fiah

Approved by:

Sudip Mittal (Major Professor)
J. Edward Swan, II
Stephen Torri
T.J. Jankun-Kelly (Graduate Coordinator)
Jason M. Keith (Dean, The Bagley College of Engineering)

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

May 2023
Copyright by

Eric Kudjoe Fiah

2023
In this thesis, different Machine learning (ML) algorithms were used in the detection of anomalies using a dataset from a Gas pipeline SCADA system which was generated by Mississippi State University’s SCADA laboratory. This work was divided into two folds: Binary Classification and Categorized classification.

In the binary classification, two attack types namely: Command injection and Response injection attacks were considered. Eight Machine Learning Classifiers were used and the results were compared. The Light GBM and Decision tree classifiers performed better than the other algorithms used.

In the categorical classification task, Seven (7) attack types in the dataset were analyzed using six different ML classifiers. The light gradient-boosting machine (LGBM) outperformed all the other classifiers in the detection of all the attack types. One other aspect of the categorized classification was the use of an autoencoder in improving the performance of all the classifiers...
used. The last part of this thesis was the use of SHAP plots to explain the features that accounted for each attack type in the dataset.

Keywords: Anomaly, Autoencoders, Machine learning, Supervisory Control And Data Acquisition
DEDICATION

Thanks be to God for giving me the strength and wisdom to complete this thesis. I dedicate this work to my wife Adjoa and daughter Soteria for their love and support.
ACKNOWLEDGEMENTS

I would like to thank Dr. Sudip Mittal for the time and expertise he gave me throughout this work. I would like to thank him for his guidance, patience, and valuable corrections in the course of this work and beyond.

I would also like to thank my committee members Dr. Ed Swan II and Dr. Stephen Torri for agreeing to serve on my committee. I would like to thank them, especially for all the time, advice, and knowledge they provided for the success of this work.
TABLE OF CONTENTS

DEDICATION ................................................................. ii

ACKNOWLEDGEMENTS .................................................. iii

LIST OF TABLES ............................................................ vi

LIST OF FIGURES .......................................................... vii

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE .............................. viii

CHAPTER

I. INTRODUCTION .......................................................... 1

1.1 Objectives, Research Questions, and Approach .................................. 3
1.2 Motivation ............................................................... 4
1.3 The architecture of SCADA systems .............................................. 5
1.4 Modbus Protocol ......................................................... 6
1.5 Organization of the Work .................................................. 7

II. RELATED WORK .......................................................... 8

2.0.1 Background ............................................................ 8
2.0.2 SCADA Security Challenges ........................................... 9
2.0.3 Some recent attacks on SCADA systems .................................. 11
2.0.3.1 Colonial gas pipeline(2021) ....................................... 12
2.0.3.2 Natural gas plant attack, US(2020) ................................ 12
2.0.3.3 EDP Power plant cyber attack(2020) ............................. 13
2.0.3.4 Australian beverage firm cyber attack(2020) ...................... 13
2.0.3.5 Water Facility Attack(2020) ....................................... 13
2.0.3.6 Norsk Hydro (2019) ................................................ 13
2.0.4 Anomaly Detection ..................................................... 14

III. METHODOLOGY .......................................................... 21
3.1 Background .................................................. 21
3.1.1 Why limited SCADA Systems Dataset .................... 23
3.1.2 Dataset Description ...................................... 25
3.1.3 Various Attack types in the Dataset ....................... 26
  3.1.3.1 Command Injection .................................... 26
  3.1.3.2 Response Injection ................................... 27
3.1.4 Features in Dataset ..................................... 27
3.2 Data Preprocessing ......................................... 29
  3.2.1 Missing data handling .................................... 29
  3.2.2 Feature Selection ...................................... 30
3.3 Modeling Stage ............................................ 30
  3.3.1 Machine Learning Algorithms ............................ 31
    3.3.1.1 Decision tree ....................................... 31
    3.3.1.2 Support Vector Machine ............................. 31
    3.3.1.3 Multilayer Perceptron (MLP) ......................... 32
    3.3.1.4 Autoencoders ........................................ 32
    3.3.1.5 Random Forest Classifier ........................... 33
    3.3.1.6 K-nearest neighbor .................................. 33
    3.3.1.7 Logistic Regression ................................. 34
  3.3.2 Evaluation Metrics ..................................... 34

IV. RESULTS AND DISCUSSION ................................. 37
  4.1 Binary Classification Performance .......................... 38
    4.1.1 Command Injection Classification ...................... 38
      4.1.1.1 Analysis of the Performances of the ML Classifiers .. 39
      4.1.1.2 Explainability of the Performance of Command injection . 42
    4.1.2 Response Injection Classification ...................... 44
      4.1.2.1 Explainability of the Performance of response injection attack ........................................ 45
  4.2 Categorical classification Performance ...................... 48
    4.2.1 Analysis of the categorized classification models .... 48
    4.2.2 Comparison of Classification reports .................. 50
    4.2.3 Improving Results using Autoencoders ................ 53
    4.2.4 Explainability of Categorized classification .......... 54

V. CONCLUSION AND FUTURE WORK .......................... 56
  5.1 Future Work ............................................... 57

REFERENCES ..................................................... 58
\begin{itemize}
\item[	extbf{2.1}] Attack types on SCADA systems \hspace{1cm} 12
\item[	extbf{3.1}] List of attack types in dataset \hspace{1cm} 26
\item[	extbf{3.2}] Features in dataset \hspace{1cm} 28
\item[	extbf{3.3}] Confusion Matrix showing evaluation Metrics \hspace{1cm} 35
\item[	extbf{3.4}] Evaluation Metrics \hspace{1cm} 36
\item[	extbf{4.1}] Binary Classification of Command Injection attack \hspace{1cm} 38
\item[	extbf{4.2}] Binary Classification of Response Injection attack \hspace{1cm} 44
\item[	extbf{4.3}] Categorized Classification of all the attack types \hspace{1cm} 49
\item[	extbf{4.4}] Decision tree model Classification report \hspace{1cm} 51
\item[	extbf{4.5}] LGBM model Classification report \hspace{1cm} 52
\item[	extbf{4.6}] Using Autoencoder to improve the scores \hspace{1cm} 53
\end{itemize}
# LIST OF FIGURES

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A Simple SCADA system topology</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Graphical Representation of Methodology</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Missing values in dataset</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Performance analysis Using AUC-ROC</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Feature Importance using Shap plots</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Performance analysis Using AUC-ROC in Response injection attack</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>Feature importance of the Response injection attack</td>
<td>47</td>
</tr>
<tr>
<td>4.5</td>
<td>Feature Importance of Categorized classification using Shap plots</td>
<td>55</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE

DoS – Denial of Service
HMI – Human Machine Interface
NMRI – Naïve Malicious Response Injection
CMRI – Complex Malicious Response Injection
MPCI – Malicious Parameter Command Injection
MSCI – Malicious State Command Injection
ML – Machine learning
MLP – Multilayer perceptron
PLC – Programmable Logic Controller
SCADA – Supervisory Control and Data Acquisition
SVM – Support vector machine
RTU – Remote Terminal Unit
RF – Random Forest
VAE – Variational autoencoder
LGBM – Light gradient-boosting machine
Supervisory Control and Data Acquisition (SCADA) systems are very useful in many industrial systems. They are deployed in industrial systems such as nuclear power generation systems, public transport, and wastewater plants in the monitoring and controlling of the operations of the systems. Until recently, little or no attention has been given to the security of SCADA systems. The primary cause of this is that these systems were typically operated with unclear methods in the past and had minimal interaction with external entities. However, the current trend is shifting towards greater connectivity, with SCADA systems adopting more conventional protocols. Additionally, the deregulation of various sectors, particularly the electricity industry, renders their control systems more susceptible to exploitation by malevolent insiders[8].

SCADA systems are made up of software program that is installed on hardware and connected to it through Programmable Logic Controllers (PLCs) or other similar commercial hardware modules. They do not only control but predominantly provide supervisory role[17]. In the past, SCADA systems used isolated networks and non-typical communication methods to prevent potential attacks. However, as time passes, these networks have become linked with business networks and the Internet. Additionally, conventional communication protocols are being adopted for SCADA systems [63]. The recent surge in interlinked devices aimed at enhancing efficiency and cutting
expenses has made SCADA systems vulnerable to numerous cyber threats, and this issue requires serious attention. SCADA systems have been extensively utilized for many years to oversee and manage various operations in Critical Infrastructures, such as the power industry, water distribution, oil refineries, and industrial processes. Some of these operations are crucial components of a nation’s infrastructure, including nuclear power generation, public transportation, and wastewater plants. As a result, an attack on such systems can have severe consequences. Currently, numerous vulnerabilities in SCADA systems and software have been identified, although the most targeted systems are still those with hosting Operating Systems.[7].

SCADA systems have been vulnerable to cyber-attacks, with the types of attacks depending on the system’s architecture and configurations. Due to the vulnerabilities of the SCADA system, attackers become successful and this can lead to losses of lives and finances and may even result in environmental catastrophes. Attacks on the Internet and business information systems frequently exploit weaknesses in communication protocols and their implementations. Similarly, SCADA systems are also vulnerable to such attacks, but there is limited knowledge about the specific vulnerabilities in SCADA protocols [10].

Traditionally, SCADA systems make use of standard ICT security measures such as firewalls and intrusion detection systems for their protection against attacks. However, anomaly detection in SCADA systems performs better than conventional information and communications technology due to their underlying distinctiveness.

Anomaly detection methods assume that something abnormal is suspicious so anomaly detection systems monitor for abnormalities in the network traffic. An anomaly detection system should
have an opinion on what should be considered an intrusion[21]. The remainder of the chapters will focus on machine learning algorithms for detecting anomalies in SCADA systems.

1.1 Objectives, Research Questions, and Approach

The objectives of this thesis are as follows:

1. To explore how to use machine learning techniques to detect security threats in SCADA systems and specifically gas pipeline systems. The goal is to improve the accuracy of detecting various security threats.

2. To evaluate and contrast several machine learning algorithms based on their capability to detect different types of attacks within SCADA systems.

With the existence of numerous anomaly detection methods in traditional SCADA systems, the significance of developing novel solutions may be questioned. The goal of this thesis is to address the following research questions:

**RQ 1:** *What machine learning techniques can be employed to identify anomalies in SCADA traffic networks?*

To answer this question, we will explore different machine-learning algorithms for detecting anomalies in SCADA networks. This comprises deep learning and traditional Machine learning methods.

**RQ 2:** *What are the most effective techniques for identifying anomalies in critical infrastructure systems?*

To answer this question, We will compare different machine learning algorithms based on their accuracy in order to identify effective anomaly detection methods within the SCADA system.
The rest of this proposal is organized as follows: Section 2 focuses on Related work, section 3 focuses on the proposed methodology and dataset, and section 4 focuses on expected results and conclusion.

1.2 Motivation

The motivation behind anomaly detection in SCADA systems stems from several recent attacks on SCADA system facilities. Some of these attacks include the colonial gas pipeline attack which occurred in 2021, the Portuguese EDP power system attacks, and the Israeli water facilities cyber attacks. There has been a rise in cyber attacks on critical infrastructure systems in recent years. These attacks can result in heavy financial loss which can cause harm to the economy. Anomaly detection can help in the early detection of these attacks which can help in safeguarding these systems from damage. Critical infrastructures are today under more attack than ever before. In addition to dealing with accidents and shifting environmental circumstances, vital infrastructures are under a lot of stress from cyberattacks because of their size, sophistication, and frequency. Strategies for protecting critical infrastructure must be updated often to address newly discovered threats.
### 1.3 The architecture of SCADA systems

The supervisory control and data acquisition (SCADA) is a combination of software and hardware components that allows industrial companies to do the following [3]:

1. Locally or remotely control industrial processes.
2. Real-time data is observed, gathered, and processed.
3. to establish direct communication with various equipment, including sensors, valves, pumps, motors, and other devices. 4. Record occurrences in a file for future reference [3].

In general, SCADA systems are made up of four (4) components (illustrated in figure 1.1) [39] namely:

1) **The Physical system**: This is the gas pipeline with embedded sensors that measure quantities
such as the pressure and density of the gas pipeline. It also contains embedded actuators to control mechanisms such as solenoids, valves, or gates.

2) **The Programmable Logic Controller (PLC)**: This refers to a computer system that is designed for automation and communication with a range of devices, including factory equipment, HMI’s, sensors, and end devices. The information collected is then transmitted to other computers that run SCADA software.

3) **Communication links**: Allows data exchange between the Programmable Logic Controllers and other computers. It enables supervisory software to query information from the plant (aka. the physical system). This Communication follows a set of rules called a protocol.

4) **Human Monitoring Interface (HMI)**: The Human Machine Interface (HMI) serves as a means for a human operator to interact with a process by presenting data gathered from RTUs and PLCs. Essentially, it is the graphical user interface (GUI) of a SCADA system, which takes in information from the Servers and transforms it into various reports such as graphs and provides process situational awareness[61].

1.4 **Modbus Protocol**

Modbus is a widely used communication protocol for serial communication among industrial devices such as PLCs, RTUs, and HMI’s. SCADA systems commonly employ this protocol as it is one of the most popular protocols [47]. The Modbus is a Simple master-slave protocol that was first intended for use with serial lines. In Modbus, the master always initiates communication by sending read or write requests to slaves. Since the slave responds to each request independently, Modbus has no concept of sessions[7]. Modicon, now known as Schneider Electric, was created
in 1979 [34]. Modbus is a popular choice due to its open accessibility and lack of royalties, as well as its ease of implementation. Additionally, it offers a flexible way to transfer data without placing excessive limitations on manufacturers, unlike other communication protocols. [35][47].

1.5 Organization of the Work

The remaining of this work is arranged in this manner: Chapter 2 focuses on some related works in SCADA systems, some recent attacks on SCADA systems, and an overview of anomaly detection. Chapter 3 provides the methodology used in this thesis. This includes the dataset description, the data preprocessing, the machine learning algorithms used, and evaluation methods. Chapter 4 discusses the results. This is organized in two folds: the first part on Binary classification anomaly detection and the second part on categorical anomaly detection. Finally, chapter 5 provides conclusions on this thesis and future works.
CHAPTER II

RELATED WORK

2.0.1 Background

The purpose of developing SCADA systems was to assist in regulating and managing critical infrastructures, such as the gas pipeline and electrical power grid systems. One of its applications includes finding faults, isolating equipment, and restoring the operation of the system [14].

The SCADA system, which is used to control and monitor industrial systems includes a computer running specialized software. The different parts of the SCADA system are linked together using different types of communication protocols. Due to the interconnectivity of SCADA systems which was not so in the past, they are subjected to various forms of cyber attacks which is a major concern [15].

SCADA systems have been used for a long time to monitor and regulate various processes in Critical Infrastructures (CI) and a successful attack can have severe consequences. Currently, SCADA systems and software have several vulnerabilities, with operating systems being the most commonly exploited.[7]. Currently, due to standardized communication protocols, SCADA systems can now be integrated with both the Internet and corporate networks. However, this new context makes SCADA systems more vulnerable to various threats because of their widespread deployment, distributed operating mode, and increasing connectivity. [26].

Previously, SCADA systems were thought to be secure due to their use of specialized hardware
and software, unique communication protocols, and isolated networks. However, modern SCADA systems have adopted standard platforms, hardware, and software, and have become more interconnected. This has resulted in ICT vulnerabilities and attack methods becoming a threat to SCADA systems, despite the fact that the standard hardware and software components are more rigorously tested and secure than ever before. One benefit of this evolution is that it has reduced implementation costs. While standard ICT security measures such as firewalls and IDS have been used to protect SCADA systems, they may not be suitable for the specific communication protocols used in industrial and critical infrastructure (CI) settings. These measures are useful for defending against common attacks such as worms, viruses, and DoS, and can detect specific attacks with known signatures. However, additional measures are necessary to identify anomalous behavior that deviates from normal CI operations [21].

2.0.2 SCADA Security Challenges

A SCADA system cyberattack could have disastrous repercussions. Public safety and health may be greatly impacted by the consistent and dependable operation of SCADA systems. As a result, if SCADA systems experience any security breaches, it could put the health and safety of the general public at risk.

A hacker may, for instance, get access to critical infrastructure (CI) systems such as gas, power and water systems and cause devastating damages by shutting them down. They could also demolish vital military facilities [48]. Hong and Lee [25] talk about the security concerns that come with SCADA and smart grid communication technologies in their paper, “Challenges and Direction toward Secure Communication in the SCADA System”. Their discussion delves into how the
recent protocols for communication in SCADA have become more susceptible to cyber-attacks due to their integration into sophisticated networks. These protocols were originally intended for isolated networks and thus lack safeguards for connectivity to larger networks, resulting in security weaknesses.

Additionally, Hong and Lee [25] talk about a few issues with intrusion detection systems. They claim that in order to detect any aberrant behavior occurring within the system, SCADA IDSs need to analyze network traffic patterns[59].

SCADA systems risk evaluation in terms of security was carried out by Cherdantseva et. al [13]. This analysis covered over twenty-four(24) areas of evaluation. They proposed a straightforward system for classifying risk assessment techniques for SCADA systems after analyzing the data. Furthermore, they identified and discussed five research obstacles in the field, as well as potential solutions.

Igure et al.[26] provides a summary of the security situation of SCADA networks. The writers covered common SCADA networks’ security risks and weaknesses. The paper presents a summary of SCADA networks, including their communication protocols, vulnerabilities, and security risks. Additionally, ongoing efforts to address these risks were discussed. The paper also highlights the work of different organizations aiming to standardize SCADA security technologies.

Nicholson et al. [45] carried out an analysis of SCADA systems security challenges. Their article reviews current research and offers a clear summary of the dangers, vulnerabilities, and ways to minimize harm related to SCADA security.

Pliatsios et al. [48] have highlighted that a major challenge for SCADA systems is the lack of advanced security technologies tailored to their specific needs. SCADA systems have lower
computing power than traditional computer systems and require unique security measures. Additionally, security features are not always a priority when designing devices or protocols for SCADA systems due to cost or computational limitations, leading to potential vulnerabilities. Because SCADA systems are linked to one another either directly or indirectly, they are subject to many of the same vulnerabilities as communication networks. Despite the common belief that many SCADA systems are isolated from other networks, it has been repeatedly proven that they can also be indirectly connected to the Internet through online maintenance tools [31]. As illustrated in Table 2.1, Pliatsios et al.[48] categorize threats to SCADA systems into a variety of categories.

Over the past few years, there have been multiple instances of cyber-attacks targeting SCADA networks. These attacks have led to considerable financial losses, harmed people, and caused environmental damage. Industrial networks can be targeted in various ways, but the most notable ones include market manipulation, insider attacks, common ransomware attacks, and finally, remote or vendor site breaches.

### 2.0.3 Some recent attacks on SCADA systems

In the past, due to the isolated nature of SCADA systems, they did not face many threats like today. The sharp rise in attacks against SCADA systems is due to their interconnectivity which was not so some years back. Instances of risks to SCADA systems consist of an incident in Maroochy Shire, Queensland, where a sewage treatment system was attacked leading to the discharge of 800,000 liters of untreated sewage into neighboring parks and rivers, which resulted in the death of aquatic creatures, foul smell, and a change in water color [54]. The following are some of the recent attacks on SCADA systems.
### Table 2.1

Attack types on SCADA systems

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Violation Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denial of Service</td>
<td>Availability</td>
</tr>
<tr>
<td>Eavesdropping</td>
<td>Authorization, confidentiality</td>
</tr>
<tr>
<td>Man-in-Middle</td>
<td>Authentication, Confidentiality, Integrity</td>
</tr>
<tr>
<td>System break-in</td>
<td>Authentication, Authorization</td>
</tr>
<tr>
<td>Virus</td>
<td>Availability, Integrity</td>
</tr>
<tr>
<td>Trojan</td>
<td>Authorization, Confidentiality</td>
</tr>
<tr>
<td>Worm</td>
<td>Confidentiality, Integrity, Authorization</td>
</tr>
</tbody>
</table>

2.0.3.1 Colonial gas pipeline (2021)

Also, on May 7, 2021, a cyberattack was launched against the American colonial gas pipeline. The gas Pipeline, which transports petroleum products between Texas and New York and also accounts for about 45% of all petroleum used on the East Coast, was the target of the cyberattack. One of their biggest pipelines for refined products has been shut down by this ransomware attack [1].

2.0.3.2 Natural gas plant attack, US (2020)

In Feb 2020, a facility belonging to natural gas company suffered a cyber attack which caused it to be shut down for two days in the US. This was reported by Homeland Security who claimed it prevented the workers from getting access to some vital information [50].
2.0.3.3 EDP Power plant cyber attack(2020)

A ransomware assault in the power generation sector resulted in the loss of around 10TB of sensitive data and subsequent blackmail for the Portuguese energy company EDP potentially leading to information exposure[64].

2.0.3.4 Australian beverage firm cyber attack(2020)

In the food manufacturing industry, Lion, a significant Australian beverage firm that produces and sells drinks, was the subject of a significant ransomware attack. The attack was carried out in stages and had a significant effect on the business’s operations in June 2020 [27] [64].

2.0.3.5 Water Facility Attack(2020)

In 2020, there has been an attack on the Israeli water Facilities. Israel’s National Cyber Directorate issued a warning to the water industry advising them to upgrade their ICS software, modify passwords on equipment with internet access, and limit their exposure to the internet, in the wake of an assault on wastewater treatment plants. The suspicious behavior that resulted from the adversaries’ targeting of Programmable Logic Controllers (PLCs) allowed for their identification [41].

2.0.3.6 Norsk Hydro (2019)

A Norwegian corporation called Norsk Hydro manufactures aluminum and produces renewable energy. An updated LockerGoga ransomware program targeted Norsk Hydro, affecting 22,000 PCs across more than 170 of their facilities [40], [41]. The manual process was resumed or the flow of
molten metal lines was stopped. Recovery from this cost the organization more than £45 million [41].

2.0.4 Anomaly Detection

SCADA systems are vital in managing large industrial facilities such as nuclear power plants, gas pipelines, wastewater treatment plants, and public transport systems. Historically, little consideration has been given to their security due to their use of obscure protocols and limited connectivity to external networks. However, this is changing as SCADA systems become more interconnected with the adoption of standard protocols, and the deregulation of industries such as electricity makes them more susceptible to tampering by malevolent insiders[8].

It is important to equip SCADA systems with strong security measures and speedy detection of cyberattacks due to the fact that their interconnected networks and information exchange features leave them vulnerable to such attacks [16]. Data patterns that deviate from a well-established definition of typical behavior are known as anomalies. Anomalies may be introduced into the data for a number of reasons, including malevolent action, such as credit card fraud, cyber-intrusion, terrorist activities, or system failure, but all of the causes share the trait of being fascinating to the analyst[11].

Numerous studies, reviews, books, and articles on anomaly detection have been published. Hodge and Austin[24] carried out a survey on techniques for outlier detection. They made a comparative analysis of statistical, neural networks and machine learning models for the detection of anomalies.
Anomaly detection using conventional methods is not reliable because they rely on a database of attack signatures. These signatures must align with specific attack types and their features. However, the drawback of this approach is that it requires the interference of humans to identify cyber-attacks which is time-consuming and resource-intensive [46]. Moreover, in modern SCADA networks, the air gap principle is no longer effective, and therefore, adversaries can frequently access the control system through the Internet [9]. Intrusion detection in SCADA systems can be grouped into signature base detection, anomaly detection, and specification-based detection.

There are several methods employed by various researchers in anomaly detection in SCADA systems. One such is the use of Autoencoders as a machine learning method in anomaly detection. Alan Preciado-Grijalva et al. employed using variation recurrent Autoencoders for anomaly detection in wind turbine time series systems [49].

Variational Recurrent Autoencoder (VRAE) was employed for learning the representation of the time series data and classification algorithms were used in the final stage of detection. The result showed a 96% accuracy detection rate on the testing dataset. Variation autoencoders were chosen for this task due to their ability to learn a low-dimensional representation of data points, reproducing the data into two-dimensional latent space [28]. Moreover, Variation autoencoders achieved "state-of-the-art" and comparable with relevant baselines in two different classification tasks on image generation [28] and data reconstruction [65].

Also, in the area of healthcare, Joao Pereira et al. used variation Recurrent autoencoders in both representation learning and detection of the anomaly by using an electrocardiogram dataset. Their method produced outstanding results by surpassing other previously used methods.
Dimension reduction techniques like PCA have been used in many cases on different datasets and they produced good results. However, when dealing with datasets that have underlying nonlinear nature, it is difficult to employ these methods. Variation autoencoders have attained promising feats in dealing with such nonlinearity with an accuracy of over 90% [66].

A reliable dataset is one of the most crucial components in the creation of a novel intrusion detection strategy because it will be used to test the method and validate the results. While industrial network traffic is sparse, home and office networks have a lot of datasets available to them. Lemay and Fernande [23] present one recent dataset. They suggest a Mobus/TCP tool- and sandbox-based architecture for traffic simulation scenarios. Additionally, they released a dataset that contained fraudulent traffic they had added [47].

Morris et. al describe four datasets for intrusion detection System(IDS) research [42]. The datasets consist of information obtained through a series of attacks on testbeds using the "Modbus application layer protocol". They argue that although there are many data sets available to train and test intrusion detection systems for conventional information technology systems, there is a lack of availability of SCADA network traffic data for the security of SCADA systems. Random forest and K-nearest neighbor are examples of machine learning algorithms utilized to identify these attacks on the simulated system [57].

In order to identify threats in real-time, Keliris et. al [32] created a "process-aware" supervised learning protection method that takes into account an ICS’s operational behavior. They evaluated various sorts of attack vectors on their hardware controllers and employed a benchmark chemical process. They were able to discriminate between disturbances and malicious activities by using the model which was trained to identify anomalies in the system.
Abhijit Kulkarni et. al [36] used a Support vector machine with the inclusion of a knowledge algorithm. This was used to solve a benchmark problem in chemical engineering called the Tennessee Eastman Process. The knowledge-incorporated approach makes use of data on the examples’ horizontal translation invariance in the tangent direction.

Mantere et. al [40] investigated a group of characteristics that could be used for an anomaly detection system in a practical network environment of industrial facilities. They used network architecture diagrams and information obtained from analyzing network traces to describe the network being studied. The network trace was obtained from an operational industrial process control network and includes both control data and details about data transfers between the control network and the office network.

Tomin et. al [58] created an automated multi-model methodology for evaluating online security. The resampling cross-validation method was employed in training multiple algorithms. Then, depending on performance, the top model among the ML algorithms was chosen and applied live. They contend that applying ML approaches can help overcome the difficulties associated with designing and running future industrial systems while maintaining security to a reasonable degree [57].

Dayu et al. introduced the two fundamental detection approaches. They include using signatures in detection and also using an intrusion detection approach. Anomaly intrusion detection approaches designed for monitoring crucial industrial systems, such as nuclear power plants, were also described. The technique is applied to a simulated SCADA system using an ”auto-associative kernel regression (AAKR) model” together with the ”statistical probability ratio test (SPRT)”. 
This method performance of these methods is good in the detection of common attacks in SCADA systems.

Perez et. al[46] assessed the effectiveness of Machine Learning (ML) for detecting unauthorized access in SCADA systems. They utilized a dataset from the Mississippi gas pipeline and implemented two classification algorithms, namely Random Forest and Support Vector Machine. Based on the precision and F1 scores, the Random Forest algorithm outperformed the SVM algorithm.

Dong and Peng [18] recommended using an SVM approach to classify attacks on a Modbus network. They employed the Wireshark software to capture and analyze data packets from an authentic Modbus device. The recorded data was parsed and sequences of function codes and register addresses were created from it. The frequency of these pattern subsequences was calculated using a combination of function codes and register addresses and then mapped to the same dimension eigenvector. This allowed for the communication characteristics of multiple packets in the Modbus TCP/IP communication process to be described by converting combinations of different lengths to the same length vector. The results of the experiment showed that the proposed SVM method had an advantage with a classification accuracy of 94.13%. [48]. Jiang et. al [29] reported the detection of anomalies through automated means from SCADA systems that are centralized, along with their associated measurements and commands in the context of SCADA-field equipment interactions. They took advantage of the one-class Support vector machine idea and controlled it to identify odd behavior in the input data. The algorithm proposed by the researchers demonstrated a high detection rate in identifying potential threats in simulation data sets obtained
from telecom networks. This presents an opportunity for further exploration and enhancement of the algorithm toward effective solutions for safeguarding SCADA systems.

C. Alcaraz, L. Cazorla, and G. Fernandez proposed anomaly detection methods as an effective way of improving the context-awareness of Smart Grid Systems [5].

Jason Stamp, John Dillinger, and William Young [55] described major vulnerabilities in critical infrastructure systems and effective ways of dealing with these issues to avoid the breakdown of these systems.

B. Phillips et. al [47] used machine learning algorithms such as Decision trees, K-means, etc. for anomaly detection on a SCADA dataset from the Mississippi gas pipeline. The dataset was examined using several machine learning algorithms, and it was discovered that Decision Trees and K-nearest Neighbors algorithms outperformed K-means clustering when analyzing the various attack types.

The dataset used in this research was a lab-stimulated dataset put together by Morris, Vaughn, and Dandass [43]. Due to the difficulty in getting datasets from industrial facilities, performing a real analysis on SCADA systems is difficult. Finding solutions to SCADA system attacks is difficult because getting datasets that are open to a public use can aid in research on the various attack types. In recent years, there have been many attacks such as the Stuxnet on the Iran nuclear plant and many more. One of the available datasets that many researchers used for intrusion detection was put together by Lemay et. al [37] which is a simulation of attacks on network traffic. The problem with this dataset as discussed, lacked a real-world scenario of attacks.

Abdulmohsen Almalawi et.al [6] proposed an unsupervised anomaly-based detection of cyber
attacks on the SCADA systems. The research was carried out on a real-world dataset and simulated
dataset and they proposed techniques for anomaly detection in SCADA systems.

Moreover, Linda et. al [38] proposed a more elaborate method of intrusion detection in critical
infrastructure systems using a neural network and backpropagation training algorithm. However,
this method requires a considerable number of time during training.

Alfonso et. al [62] conducted anomaly detection on ”digital control” systems that are commonly
used in SCADA systems. They investigated two techniques for detecting anomalies: ”pattern-
based detection” for identifying irregular communication patterns and ”flow-based detection” for
detecting atypical traffic patterns of individual flows. The experiments were successful in detecting
attacks on MODBUS servers. The researchers concluded that the predictability of communication
patterns in SCADA systems aided in the identification of anomalous patterns. Also, IgorNai Fovino
et. al [20] performed an experimental method of identifying malicious attacks on SCADA systems
with specific-correlated process parameters.
CHAPTER III
METHODOLOGY

3.1 Background

Anomaly detection is a critical component of security in SCADA systems. It involves identifying abnormal behavior or patterns in the system that could indicate a cyberattack or other security threat. Machine learning algorithms can be used to automate the process of anomaly detection, improving the accuracy and speed of detection. Anomaly detection in SCADA systems using machine learning involves collecting and processing data from the system, training a machine learning algorithm to detect abnormal behavior, and continuously monitoring the system for any security threats. This approach can improve the accuracy and speed of anomaly detection and help protect SCADA systems from cyberattacks.

SCADA networks commonly employ command response protocols for communication. By analyzing the network communication packets and data, one can directly obtain attack signatures and thereby, detect and recognize anomalies [33]. As a result of the connectivity of SCADA systems to the internet, there has been an increase in attacks on these systems. Some common types of attacks on SCADA Systems include the following:

1. Malware: Malware can be used to gain unauthorized access to the SCADA system or to steal sensitive data. Malware can also be used to disable the SCADA system, causing disruptions to industrial processes.

2. Denial-of-Service (DoS) attacks: DoS attacks can overload the SCADA system with requests,
causing it to become unresponsive and potentially disrupting industrial processes.

3. Man-in-the-middle (MitM) attacks: MitM attacks can intercept and modify data between the SCADA system and its connected devices, allowing an attacker to manipulate the industrial processes being monitored and controlled by the system.

4. SQL injection attacks: SQL injection attacks can exploit vulnerabilities in the SCADA system’s database, potentially allowing an attacker to gain unauthorized access to sensitive information.

This chapter is divided into the following sections: Dataset description, Attack types, Data preprocessing, Feature selection, Machine learning algorithms, and Performance metrics. figure 3.1 on page 24 shows the graphical representation of the methodology used in this study.
3.1.1 Why limited SCADA Systems Dataset

Most SCADA systems are owned by individual businesses and some by State organizations. There are a limited amount of real-world data available due to companies affected inability to provide datasets. Also, because SCADA systems form a sensitive part of industrial systems, sharing their dataset will further expose the system to other attacks. These are some reasons why there are limited available datasets. Researchers carry out analysis on synthetically generated datasets such as the Mississippi gas pipeline dataset. figure 3.1 on the following page shows the graphical representation of the processes used in this project. The methodology is divided into the following steps using the available dataset. The process includes the description of the dataset, data preprocessing, feature engineering, selection of machine learning algorithms, training of models, and, evaluation of the model. The remainder of this chapter will describe the following processes in detail.
Figure 3.1

Graphical Representation of Methodology
3.1.2 Dataset Description

The SCADA gas pipeline dataset is a collection of data related to the operation of a gas pipeline system. The SCADA gas pipeline dataset is used to analyze the performance of the pipeline system and detect any abnormalities or potential issues. The data can be used to create predictive models that can forecast pipeline performance and help prevent pipeline failures or accidents. Wei Gao et. al [42] worked on the first dataset for a gas pipeline system but it was later found out the dataset contains easily identifiable patterns making it easier for machine learning algorithms to score up to 100% in accuracy [60].

In this study, a dataset introduced by Morris, Thornton, and Turnipseed [44], which consists of a computer-simulated SCADA network responsible for transporting petroleum products through a gas pipeline to be sold on the market was utilized for the studies. The potential consequences of cyber attacks on the control systems that oversee gas pipelines can be catastrophic [47]. This can bring physical damage to the gas pipeline and cause financial loss. The website known as the Industrial Control System (ICS) Cyber Attack [2] is where the SCADA dataset used for this research can be found.

The dataset was created using the SCADA lab located at Mississippi State University (MSU). The dataset consists of a total of 274,628 instances. The dataset is divided into rows, each with several columns, or "features," as they are sometimes referred to. Twenty features are present in the dataset. A detailed description of the dataset and the methods employed in generating it can be found here [60].
3.1.3 Various Attack types in the Dataset

Table 3.1

<table>
<thead>
<tr>
<th>Attack types</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal(0)</td>
</tr>
<tr>
<td>Naive Malicious response Injection</td>
<td>NMRI(1)</td>
</tr>
<tr>
<td>Complex Malicious Response Injection</td>
<td>CMRI(2)</td>
</tr>
<tr>
<td>Malicious State Command Injection</td>
<td>MSCI(3)</td>
</tr>
<tr>
<td>Malicious Parameter Command Injection</td>
<td>MPCI(4)</td>
</tr>
<tr>
<td>Malicious Function Code Injection</td>
<td>MFCI(5)</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>DoS(6)</td>
</tr>
<tr>
<td>Reconnaissance</td>
<td>Recon(7)</td>
</tr>
</tbody>
</table>

There are a total of seven(7) different types of attacks in the dataset. The various attack types are listed in Table 3.1 which were first identified in a study by Gao[42]. The four main categories into which these attacks can further be divided are command injection, response injection, denial of service (DoS), and reconnaissance (Reconn) attacks.

3.1.3.1 Command Injection

In a command injection attack, the attack scheme is to supply unauthorized instructions to the gas pipeline control system. The attacker can exploit this vulnerability to execute arbitrary
commands on the system, gain unauthorized access, or perform other malicious activities. This 
attack is aimed at causing the system to behave abnormally. In the dataset, these attack groups are 
MSCI, MPCI, and MFCI.

3.1.3.2 Response Injection

There are two types of behaviors associated with attacks that utilize response injection. The first 
type is called naive malicious response injection (NMRI), which involves abnormal and improper 
behavior that deviates from the system’s normal operations. Such attacks are usually carried out 
by hostile attackers who lack knowledge of how the physical system works. The second type is 
called complex malicious response injection (CMRI), which involves the use of knowledge about 
the system’s state and physical processes to create patterns that resemble common behaviors [60].

3.1.4 Features in Dataset

The dataset comprises twenty (20) features as stated in Table 3.2. A detailed description of 
all features can be found here [60]. There are a total of three dependent(target) columns namely: 
binary, specific, and categorized. The remaining seventeen (17) columns are independent columns.
Table 3.2

Features in dataset

<table>
<thead>
<tr>
<th>Features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>control scheme</td>
</tr>
<tr>
<td>function</td>
<td>pump</td>
</tr>
<tr>
<td>length</td>
<td>solenoid</td>
</tr>
<tr>
<td>setpoint</td>
<td>pressure</td>
</tr>
<tr>
<td>gain</td>
<td>crc rate</td>
</tr>
<tr>
<td>reset rate</td>
<td>command response</td>
</tr>
<tr>
<td>deadband</td>
<td>time</td>
</tr>
<tr>
<td>cycle time</td>
<td>binary</td>
</tr>
<tr>
<td>rate</td>
<td>categorized</td>
</tr>
<tr>
<td>system mode</td>
<td>specific</td>
</tr>
</tbody>
</table>
3.2 Data Preprocessing

Before applying machine learning algorithms to data, it is necessary to convert the data into a format suitable for analysis by the chosen machine learning technique. The effectiveness of the whole system is directly influenced by the pre-processing procedures used. Pre-processing typically consists of feature selection, data normalization, and data conversion techniques[4]. The following methods were carried out in the preprocessing stage.

3.2.1 Missing data handling

There were missing values in the feature columns in the dataset. There was a total of 2311020 missing data points in the dataset. Multiple Imputation by Chained Equations (MICE) was used in filling in the missing values in the dataset. Figure 3.2 shows the missing values denoted by a question mark(?) in the dataset before the preprocessing process.
3.2.2 Feature Selection

In the feature selection stage, various statistical techniques were used to determine the relevant features. All the categorical columns were dropped. Two methods used for feature selection were the Pearson correlation matrix and the Decision tree classifier feature importance methods. These methods produced similar results where all the features were relevant to the modeling process.

3.3 Modeling Stage

In the modeling stage, various actions were used in building the models. The dataset was divided into training and testing sets. 70% of the dataset was used for training and 30% was used for testing. The Minmaxscaler was used to normalize the data between 0 and 1 before training. This helps to make the data more uniform and easier to compare. The dataset was pre-processed and normalized before the models were created because some classifiers perform better on normalized data.

To create a method for detecting anomalies, there are several steps involved. Initially, a prediction model and its corresponding hyperparameters are chosen, and the optimal ones are identified. Subsequently, the model is trained, and a threshold is established to distinguish between samples that are representative of typical system behavior and those that indicate a cyber attack. Finally, the accuracy of the approach is evaluated through performance metrics such as precision, recall, and F1-score.
3.3.1 Machine Learning Algorithms

This section will focus on the various machine-learning algorithms that were used in this research. All techniques and algorithms that allow computers to automatically learn from huge datasets by applying mathematical models are together referred to as machine learning (ML) [4].

3.3.1.1 Decision tree

A decision tree classifier is a type of supervised machine learning algorithm that builds a tree-like model of decisions and their possible consequences. The model is constructed by recursively partitioning the input space into smaller regions, where each partition is associated with a decision or a class label. The process of building the decision tree involves selecting the most informative attribute to split the data at each internal node. A decision tree serves as a classifier by partitioning the instance space recursively. The nodes of the decision tree form a rooted tree, where the root node has no incoming edges and each of the other nodes has only one incoming edge. Internal nodes are also called test nodes, and they have outgoing edges, while the remaining nodes are called leaf nodes or terminal nodes. The internal nodes of the decision tree partition the instance space into two or more sub-spaces using a discrete function that depends on the input attribute values. Typically, each test considers a single attribute and divides the instance space based on the attribute value. If the attribute is numeric, the test condition specifies a range of values.[51].

3.3.1.2 Support Vector Machine

SVM, a supervised machine learning method, relies on the idea of a hyperplane that maximizes the margin of separation between classes in feature space with n dimensions. Both linear and nonlinear problems can be solved with it. Kernel functions are used to solve nonlinear issues.
A low-dimensional input vector is first intended to be mapped using the kernel function into a feature space that is high-dimensional. The next step is to create an optimal maximum marginal hyper-plane, which uses the support vectors to create a decision boundary[12]. To distinguish between different classes of observations, SVMs use hyperplanes in multidimensional space.

### 3.3.1.3 Multilayer Perceptron (MLP)

The multilayer perceptron (MLP) is a type of artificial neural network that consists of multiple layers of interconnected nodes, which are designed to process input data and produce output values. It consists of the connections between neurons (nodes), which are the processing units, and other components. The arrangement of these nodes involves an input layer, multiple hidden layers, and an output layer. The backpropagation technique is utilized for the ANN’s learning process. An ANN technique’s primary benefit is its capacity for nonlinear modeling via learning from larger datasets[53][4].

### 3.3.1.4 Autoencoders

A deep learning model called an autoencoder (AE) is composed mostly of a decoder and an encoder. An AE discovers the most effective way to encode input data into latent space to reproduce it. It operates under the premise that by learning the best qualities, the output can be as closely matched to the input as feasible. It has identical input and output layers, whereas the hidden layers’ dimensions are often less than the input layers. AE employs an encoder-decoder algorithm and is symmetric [22]. Sakurada and Yairi[52] suggest using autoencoders with nonlinear dimensionality reduction for anomaly detection. Both synthetic and actual data were subjected to dimensionality reduction
using an autoencoder, and their properties were clarified by comparing them to linear PCA and kernel PCA. The fake data is produced by the Lorenz system, and the actual data is the telemetry data from the spacecraft. They proved that autoencoders are capable of detecting subtle anomalies that linear PCA is incapable of. Additionally, by using denoising autoencoders, autoencoders can improve their accuracy. Additionally, autoencoders can be helpful as nonlinear approaches without requiring complicated computations like kernel PCA. Additionally, they looked at the learned features in the autoencoders’ hidden layer and demonstrated that these algorithms properly teach autoencoders how to operate in a normal state.

3.3.1.5 Random Forest Classifier

The Random forest (RF) classifier uses a decision tree in classification, and this ensemble learning technique is used to characterize the data. To categorize fresh cases, the ensemble technique integrates the indicators from various trained classifiers. A classifier that consists of classifiers structured into trees is called a random forest. The independent random vectors are distributed similarly in those classifiers. Furthermore, a single vote is contributed by each tree in favor of the most popular class. The creation of a tree is achieved through the training test, while a new random vector is not correlated with any previous random vectors that have the same distribution [56].

3.3.1.6 K-nearest neighbor

K-nearest neighbor (KNN) is a simple but effective algorithm used in classification and regression tasks. It is a type of instance-based learning, where the algorithm predicts the class or value of a new data point based on the classes or values of its k-nearest neighbors in the training data.
K-nearest neighbor (KNN) is a type of lazy learning algorithm that is non-parametric in nature. Unlike other algorithms, KNN does not perform any training when provided with the training data. This is why it is also known as a lazy learner. During the training period, KNN simply stores the data and does not make any calculations. It only creates a model when a dataset query is run.[30].

**3.3.1.7 Logistic Regression**

Logistic regression is a type of regression analysis used to model the relationship between a binary dependent variable (i.e., a variable that can take only two values, such as 0 or 1) and one or more independent variables (also known as predictors or features). The logistic regression model is commonly used for binary outcomes but can also be used for multi-class classification problems. Logistic regression is a popular method for classification problems, including in cyber security where detecting attacks is often a classification problem [19].

**3.3.2 Evaluation Metrics**

The machine learning models were evaluated using different performance metrics. The choice of these metrics depends on the binary and multi-classification task. 3.4 shows the evaluation metrics employed in analyzing the various ML models. The confusion Matrix was used in computing these metrics. Table 3.3 shows a detailed assessment of the confusion matrix. The following is the general definition of the metrics used.

**True Positives (TP):** These refer to the positive predictions that are accurate, indicating that the actual class value is 'yes' and the predicted class value is also 'yes'.

**True Negatives (TN):** These are the accurately predicted negative values, indicating that the actual class value is 'no' and the predicted class value is also 'no'.

34
Table 3.3

Confusion Matrix showing evaluation Metrics

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>Negative</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

False positives and false negatives, these values occur when your actual class contradicts the predicted class.

**False Positives (FP):** A false positive refers to a situation where the model makes an incorrect prediction that falls into the positive category.

**False Negatives (FN):** A false negative is when the model makes an erroneous prediction that belongs to the negative category.

**Accuracy:** Accuracy is a measure that indicates the proportion of data instances that have been classified correctly, calculated by dividing the number of accurately classified instances by the total number of instances.

**Precision:** Precision refers to the proportion of accurately predicted positive outcomes among all the predicted positive outcomes.

**Recall (Sensitivity):** Recall is a measure of the proportion of accurately predicted positive outcomes in the actual class, considering all observations.

**F1 score:** The F1 Score is a measure that considers both Precision and Recall by taking their weighted average. It takes into account both false positives and false negatives.

**AUC-ROC curve:** The AUC-ROC curve is a useful tool for evaluating the performance of
a classification model at different threshold levels. AUC refers to the degree of separability or distinguishability, while ROC is a probability curve that indicates how well the model can distinguish between classes. The higher the AUC, the more accurately the model correctly identifies 0 classes as 0 and 1 class as 1.

Table 3.4

<table>
<thead>
<tr>
<th>Performance Metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>$\frac{TP+TN}{TP+TN+FP+FN}$</td>
</tr>
<tr>
<td>Recall</td>
<td>$\frac{TP}{TP+FN}$</td>
</tr>
<tr>
<td>Precision</td>
<td>$\frac{TP}{TP+FP}$</td>
</tr>
<tr>
<td>F1-Score</td>
<td>$\frac{2TP}{2TP+FP+FN}$</td>
</tr>
<tr>
<td>Specificity</td>
<td>$\frac{TN}{TN+FP}$</td>
</tr>
</tbody>
</table>
CHAPTER IV
RESULTS AND DISCUSSION

In this section, the performance of different ML classifiers was analyzed in terms of their performance on the training and testing dataset. Some evaluation metrics used are accuracy, recall, Precision, F1-score, and speed. The ML algorithms used in the analysis are Random Forest, Logistic regression, Support Vector Machine, K-nearest neighbor, Multilayer perceptron, Decision tree, LightGBM, and Autoencoders. These classifiers were compared for the best machine learning algorithms for detecting the various types of anomalies.

The remainder of this section will focus on binary classification where two main attacks on the SCADA gas pipeline dataset were considered. These are command injection and Response injection attacks. These attacks were analyzed using different machine learning algorithms for the detection of these anomalies. The second part of this chapter will focus on the Categorical classification of the attacks on the dataset. The categories of attacks in the dataset include naïve Malicious Response Injection, Complex Malicious Response Injection, Malicious Parameter Command Injection, Malicious Function Code Injection, Denial of Service, and Reconnaissance attacks. These attacks can further be classified into command injection, response injection, and Functional code attacks. The scope of this work focused on Command injection and Response injection attacks.
4.1 Binary Classification Performance

The binary classification task involves classifying the data points into normal(0) and attacks(1). The dataset was divided into Command injection and Response injection attacks based on the categories they belong to.

4.1.1 Command Injection Classification

Table 4.1

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-Score</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>90.23%</td>
<td>90.23%</td>
<td>91.22%</td>
<td>87.69%</td>
<td>1.4s</td>
</tr>
<tr>
<td>KNN</td>
<td>99.06%</td>
<td>99.06%</td>
<td>99.05%</td>
<td>99.05%</td>
<td>7.2s</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>99.36%</td>
<td>99.36%</td>
<td>99.36%</td>
<td>99.36%</td>
<td>6.1s</td>
</tr>
<tr>
<td>Random Forest</td>
<td>99.30%</td>
<td>99.30%</td>
<td>99.30%</td>
<td>99.30%</td>
<td>6.0</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>90.43%</td>
<td>90.43%</td>
<td>91.18%</td>
<td>88.08%</td>
<td>21.3s</td>
</tr>
<tr>
<td>LSTM</td>
<td>90.52%</td>
<td>90.52%</td>
<td>90.52%</td>
<td>90.52%</td>
<td>117.3s</td>
</tr>
<tr>
<td>Light GBM</td>
<td>98.89%</td>
<td>98.89%</td>
<td>98.91%</td>
<td>98.87%</td>
<td>0.8s</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>90.39%</td>
<td>90.39%</td>
<td>91.35%</td>
<td>87.96%</td>
<td>4008.4s</td>
</tr>
</tbody>
</table>
4.1.1.1 Analysis of the Performances of the ML Classifiers

Table 4.2 shows the performance of all the machine learning classifiers used in command injection attack detection. The classifiers performed well in the detection of the command injection attacks.

In terms of Accuracies, the Decision tree model performed best with an accuracy score of 99.33%. The Random Forest classifier is next with an accuracy score of 99.30%. Other classifiers such as K-nearest neighbor(KNN), Light GBM, and multilayer perceptron also scored high in their respective accuracies.

Moreover, Regarding Recall performance, The KNN had a score of 99.06%, The Light GBM(LGBM) obtained a score of 98.89 %, The Random forest obtained a score of 99.30%, and the decision tree obtained a score of 99.36%. With regard to the F1 Scores, the Decision tree model performed best with a score of 99.36%. Other classifiers such as the Decision tree model, KNN, and LGBM also performed well with an average of 99% Recall. The Decision tree model performed best in the precision scores of 99.36%.

One important metric considered here is the speed in detecting these anomalies. This was computed in terms of the average time for training and prediction. The overall time shows how fast the model performed in classifying attacks on the SCADA system. Due to the critical nature of SCADA systems, early detection of attacks is essential in the protection of the system against losses. The LGBM model is the fastest with a speed of 0.8 seconds in the detection of these attacks.

In General, The Decision tree model performed best in the detection of command injection attacks. The random forest, LGBM, KNN, and MLP also can help in the early detection of this attack type. The SVM classifier performed poorly with a total time of 4008.4 seconds in command
The performance of best-performing classifiers using AUC-ROC curves. The higher the score, the better the model is at separating between the normal and attack data points. From the scores, the Random forest, LGBMClassifier, KNN, and Decision tree model performed better with a scores average of 0.99. The performances of the various classifiers show their ability to detect command response attacks on SCADA systems. The choice of one particular classifier would depend on underlying factors such as the speed of detection and the complexity of the system involved. Due to the imbalanced nature of the dataset, the metrics considered here are the F1-score, precision, Recall, and AUC-ROC scores.
Figure 4.1

Performance analysis Using AUC-ROC
4.1.1.2 Explainability of the Performance of Command injection

The feature importance plotted using SHAP is shown in figure 4.2 on the next page. From the plot, the features that accounted for the command injection attacks were the reset rate, the function code, gain, deadband, cycle time, setpoint, and system mode. These features are predominantly manipulated by the attacker by sending wrong commands to change the behavior of the system from normal functioning.
Figure 4.2

Feature Importance using Shap plots
4.1.2 Response Injection Classification

Table 4.2

Binary Classification of Response Injection attack

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-Score</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>91.28%</td>
<td>91.28%</td>
<td>92.04%</td>
<td>87.22%</td>
<td>1.1s</td>
</tr>
<tr>
<td>KNN</td>
<td>94.90%</td>
<td>94.90%</td>
<td>94.60%</td>
<td>94.70%</td>
<td>6.9s</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>95.02%</td>
<td>95.02%</td>
<td>94.88%</td>
<td>94.94%</td>
<td>0.4s</td>
</tr>
<tr>
<td>Random Forest</td>
<td>93.96%</td>
<td>93.96%</td>
<td>93.62%</td>
<td>93.75%</td>
<td>3.7s</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>91.32%</td>
<td>91.32%</td>
<td>92.08%</td>
<td>87.33%</td>
<td>2.8s</td>
</tr>
<tr>
<td>LSTM</td>
<td>91.32%</td>
<td>91.32%</td>
<td>91.32%</td>
<td>91.32%</td>
<td>135.1s</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>91.27%</td>
<td>91.27%</td>
<td>92.04%</td>
<td>87.21%</td>
<td>584.3s</td>
</tr>
</tbody>
</table>

Response injection attacks involve manipulating the information sent from the server to the client, with the purpose of providing misleading information about the system’s state. These attacks are categorized into two types: naive malicious response injection (NMRI) attacks and complex malicious response injection (CMRI) attacks [42]. The performance of the Machine learning classifiers in identifying response injection is shown in table 4.1. Using the Accuracy scores, the LGBM classifier performed best with an accuracy score of 99.99%. The Decision tree classifier, KNN, Random forest, MLP, LSTM, Logistic regression, and SVM all performed well in their scores. With regards to the precision scores, the LGBM also had the best performance of
99.99% followed by the decision tree model which scored 94.88%. All the classifiers performed better in the F1-score and the precision scores.

The decision tree model and LGBM had relatively lesser time in the training and prediction stages. Overall, all the classifiers performed well and can help in the detection of response injection attacks. The SVM classifier required much time in training and prediction therefore is not the best for this task. The LGBM classifier performed best in all the evaluation metrics and can help in the early detection of response injection attacks on the SCADA system. The performance of the various machine learning classifiers regarding their respective AUC-ROC scores is shown in figure 4.3 on the following page. The LGBM classifier obtained the best score of 1.0 followed closely by the Random forest and Decision tree classifiers. This performance shows that the classifiers are best at differentiating between normal and malicious signals. The best classifier in this task is the LGBM which scores higher in all the evaluation metrics than all the other classifiers considered.

4.1.2.1 Explainability of the Performance of response injection attack

The features contributing importantly to the response injection attack are shown in figure 4.4 on page 47. It is important to note that, the command and function codes contributed massively to the detection process. Other features such as the setpoint, pump state, gain, pressure, and control state were also important factors in response injection attacks.
Figure 4.3

Performance analysis Using AUC-ROC in Response injection attack
Feature importance of the Response injection attack
4.2 Categorical classification Performance

This section focuses on the detection of anomalies in the various categories of attacks in the dataset. The attack types were divided into four main categories namely: the command injection, Response injection, Denial of service, and Reconnaissance attack. These attack types are further divided into seven main groups which are the NMRI(1), CMRI(2), MSCI(3), MPCI(4), MFCI(5), DOS(6) Recon(7). Different machine learning classifiers were employed in the detection of these attacks in the dataset. The main classifiers considered in this study were the Random forest, light gradient-boosting machine(LGBM), K-nearest neighbor(KNN), Multilayer Perceptron, Logistic regression, and Decision tree classifier. These classifiers were compared in terms of their ability to detect the attack types from the normal(0) dataset. The evaluation metrics used in the categorized attack types classification are the precision score, F1-score, Recall, and accuracy scores. The rest of this section would focus on the analysis of the performance of the ML algorithms considered in the detection of the various attack types.

4.2.1 Analysis of the categorized classification models

The Performance of the different machine learning classifiers used in the detection of the categorized attack type is shown in Table 4.3. From the table, it can be observed that the best-performing model was the light boosting-gradient machine(LGBM). It has average accuracy and recall of 98.77%, precision, and F1-score of 98.79% and 98.74% respectively. This shows the ability of the classifier to detect most attack types in the gas pipeline SCADA system dataset. Other classifiers such as the Decision tree, KNN and Random forest performed well in their scores. The
worst-performing model is the logistic regression model and it indicates its unsuitability for this task.

Regarding the average time for performing the classification task, the Decision tree model took a total time of 0.6s in the training and prediction stage. The decision tree model and the light GBM were the best compared with the other classifiers used in the analysis.

Table 4.3
Categorized Classification of all the attack types

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-Score</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light GBM</td>
<td>98.77%</td>
<td>98.77%</td>
<td>98.79%</td>
<td>98.74%</td>
<td>29s</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>82.27%</td>
<td>82.27%</td>
<td>79.80%</td>
<td>75.67%</td>
<td>7.5s</td>
</tr>
<tr>
<td>KNN</td>
<td>94.20%</td>
<td>94.20%</td>
<td>93.76%</td>
<td>93.90%</td>
<td>62.5s</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>93.92%</td>
<td>93.92%</td>
<td>93.68%</td>
<td>93.79%</td>
<td>0.9s</td>
</tr>
<tr>
<td>Random Forest</td>
<td>92.42%</td>
<td>92.42%</td>
<td>91.83%</td>
<td>89.84%</td>
<td>23.6s</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>83.26%</td>
<td>83.26%</td>
<td>80.41%</td>
<td>77.09%</td>
<td>105.1s</td>
</tr>
</tbody>
</table>
4.2.2 Comparison of Classification reports

The performance of two main classifiers in detecting the attack types is considered in this section. The two classifiers are the Decision tree Model and the LGBM classifier. Their performance was evaluated and compared using the evaluation metrics.

The classification report of the Decision tree Model is shown in Table 4.4. It is important to note that, the classifier performed well in detecting command injection attacks which shows in its performance in classifying attack labels MSCI, MPCI, and MFCI. Also, it can be observed, it performed best in the detection of the Denial of Service(DoS) attack type and Reconnaissance(Reconn) attack type.

The performance of the Decision tree model in detecting Response injection attack types was not satisfactory. It can be seen in the precision, Recall, and F1-score of NMRI and CMRI attack types. This may be due to the nature of the CMRI attack type which is created with the aim to mimic certain normal behaviors therefore the model could not detect all and may assume they were normal behaviors. The low recall and F1-score of the NMRI attack were due to the sporadic nature of the attack type making the model finding it difficult to identify them.

Comparing the performance with the light gradient-boosting machine(LGBM) as shown in Table 4.5, it can be observed that LGBM was able to detect all attack types scoring 99% in its accuracy. This classifier performed best in the detection of Response injection attack types with an F1-score of 99% for NMRI and CMRI attacks. This shows LGBM is the best model among the other models considered in this study and outperformed all the other classifiers.
Table 4.4
Decision tree model Classification report

<table>
<thead>
<tr>
<th>Decision tree classifier</th>
<th>Attack category</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal(0)</td>
<td>96%</td>
<td>97%</td>
<td>97%</td>
</tr>
<tr>
<td></td>
<td>NMRI(1)</td>
<td>47%</td>
<td>43%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>CMRI(2)</td>
<td>63%</td>
<td>59%</td>
<td>61%</td>
</tr>
<tr>
<td></td>
<td>MSCI(3)</td>
<td>97%</td>
<td>96%</td>
<td>96%</td>
</tr>
<tr>
<td></td>
<td>MPCI(4)</td>
<td>97%</td>
<td>96%</td>
<td>96%</td>
</tr>
<tr>
<td></td>
<td>MFCI(5)</td>
<td>99%</td>
<td>96%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>Dos(6)</td>
<td>98%</td>
<td>99%</td>
<td>97%</td>
</tr>
<tr>
<td></td>
<td>Recon(7)</td>
<td>99%</td>
<td>96%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>Weighted</td>
<td>94%</td>
<td>94%</td>
<td>94%</td>
</tr>
</tbody>
</table>
Table 4.5

LGBM model Classification report

<table>
<thead>
<tr>
<th>LGBM Classifier</th>
<th>Attack category</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal(0)</td>
<td>99%</td>
<td>100%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>NMRI(1)</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>CMRI(2)</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>MSCI(3)</td>
<td>99%</td>
<td>85%</td>
<td>91%</td>
</tr>
<tr>
<td></td>
<td>MPCI(4)</td>
<td>99%</td>
<td>91%</td>
<td>95%</td>
</tr>
<tr>
<td></td>
<td>MFCI(5)</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>Dos(6)</td>
<td>99%</td>
<td>95%</td>
<td>98%</td>
</tr>
<tr>
<td></td>
<td>Recon(7)</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td>Weighted</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
</tr>
</tbody>
</table>
4.2.3 Improving Results using Autoencoders

The results of the underperforming classifiers can be improved using autoencoders. As shown in Table 4.6 The performance of all the classifiers improved by fitting the encoded data into the classifiers. The result shows a significant improvement in the ability of the classifier to detect anomalies in the dataset.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-Score</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Light GBM</strong></td>
<td>98.77%</td>
<td>98.77%</td>
<td>98.79%</td>
<td>98.74%</td>
<td>29.9s</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td>89.647%</td>
<td>99.64%</td>
<td>99.64%</td>
<td>99.64%</td>
<td>1.2s</td>
</tr>
<tr>
<td><strong>KNN</strong></td>
<td>99.77%</td>
<td>99.77%</td>
<td>99.77%</td>
<td>99.77%</td>
<td>6.2s</td>
</tr>
<tr>
<td><strong>Decision Tree</strong></td>
<td>99.79%</td>
<td>99.79%</td>
<td>99.79%</td>
<td>99.79%</td>
<td>1.5s</td>
</tr>
<tr>
<td><strong>Random Forest</strong></td>
<td>99.82%</td>
<td>99.82%</td>
<td>99.82%</td>
<td>99.82%</td>
<td>4.7s</td>
</tr>
<tr>
<td><strong>Multilayer Perceptron</strong></td>
<td>98.81%</td>
<td>98.81%</td>
<td>98.87%</td>
<td>98.84%</td>
<td>15.2s</td>
</tr>
</tbody>
</table>
4.2.4 Explainability of Categorized classification

The features that contributed majorly to the different attack types are shown in figure 4.5 on the following page. From this figure, cyclic redundancy code (CRC) contributed majorly to the class 6 attack. This attack is aimed to deplete resources and is targeted at system programs and communication links. An invalid cyclic redundancy code (CRC) attack floods a network with numerous MODBUS packets that have inaccurate CRC values [44].

On the other hand, the response injection attack type which is indicated by class 1 and class 2 is largely caused by attacks on deadband, function code, reset, system, pressure, cycle, and CRC.

The command injection attack types may be due to attacks on the features such as the control, function code, reset, setpoint, command, and length. This attack is aimed at injecting false control and changing the behavior of the system. The reconnaissance attack type indicated by class 7 attack is largely contributed by the address, length, function code, and system mode. Reconnaissance attacks have the objective of gathering information about a system either by passive means or by coercing a device to disclose information [60].
Figure 4.5

Feature Importance of Categorized classification using Shap plots
CHAPTER V
CONCLUSION AND FUTURE WORK

The analysis of this research was in two folds: 1) The detection of anomalies from the binary label. In this, the dataset was divided into two main attack types which were the Command injection (MSCI, MPCI, MFCI) and the Response injection (CMRI and NMRI) attack types. These were analyzed using different machine learning classifiers with the aim of detecting anomalies in the SCADA gas pipeline dataset. From the results and analysis in Chapter 4, it can be observed that all the models presented in this research performed well in anomaly detection in SCADA systems.

On the binary classification task, the Decision tree classifier, Light gradient-boosting machine (LGBM), Random Forest, K-nearest neighbor, multilayer perception, LSTM, and SVM were all used in the various task. In anomaly detection in the command injection dataset, the Decision tree, LGBM, and Random forest outperformed the other classifiers comparatively. In the response injection attacks, the LGBM outperformed all the other classifiers in the detection of this attack type though all the other classifiers performed well with exception of the logistic regression model.

Moreover, in the categorical classification, the dataset was analyzed based on the seven(7) main target categories and namely: the normal traffics, command injection, response injection, Denial of service attack, and Reconnaissance attack types. The focus of this thesis was on response injection and command injection attack types. The categorical anomaly detection task was divided into two
folds and the performances was analyzed. The first part was using six (6) classifiers namely; LGBM, KNN, logistic regression, Decision tree, Random forest, and MLP in the anomaly detection task. Most of the models struggled with complex malicious Response Injection (CMRI) attacks due to the nature of these attacks. CMRI is made to mimic normal behaviors therefore the model interprets them as normal instances. The LGBM and Decision tree model outperformed all the other classifiers. One important thing to note is the LGBM was able to detect the CMRI and NMRI injections in the dataset with an F1-score of 99%.

The second part focused on using autoencoders in improving the performance of the models. It was observed that all the models considered improved in their performance when encoded data points were fit into the classifiers. In general, the classifiers can be recommended for anomaly detection in the SCADA system. LGBM, Decision trees, and using autoencoders can improve the accuracy and precision of anomaly detection in SCADA systems.

5.1 Future Work

In the future, I would like to experiment with real-world datasets. Currently, getting this dataset is difficult due to the policies of most industries that use SCADA systems. This will offer me the opportunity to explore and improve on all the machine learning algorithms used in this research. Also, I would like to use explainable AI in will help to give clarity to most parts of this work. Though SHAP plots were used with the aim of explainability, more is needed to give clarity to the machine learning algorithms. Finally, I would like to publish my findings in a journal for easy access.
REFERENCES


[34] M. Klein, “MODBUS TCP/IP”.


