Detection and Mitigation Strategies for Cyber-attacks in Offshore Oil and Gas Industrial Networks

A thesis submitted in partial fulfilment

of the requirement for the degree of Doctor of Philosophy by

Abubakar Sadiq Mohammed

CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD

School of Computer Science and Informatics

Cardiff University

2024

Abstract

Industrial Cyber-Physical Systems (ICPS) increasingly rely on insecure protocols, raising security concerns in oil and gas (OG) operations. Replacing these protocols is often too expensive, highlighting the need for efficient cyber-attack detection. This thesis addresses this critical challenge by proposing a novel unsupervised anomaly detection model attack detection in OG environments.

Existing Intrusion Detection Systems (IDS) for industrial networks, primarily Machine Learning (ML)-based, often suffer from high false positive rates and limited focus on OG environments. This potentially hinders real-world adoption. To address this gap, we introduce the Sliding Time-window Anomaly Detection (STADe) model – a novel approach that leverages the inherent periodicity of industrial network traffic for anomaly detection.

The STADe model segments network packet inter-arrival times into time windows and analyzes periodicity within each window. This approach demonstrably reduces False Discovery Rates (FDR) compared to existing methods.

Experiments evaluate existing ML-based IDSs and leverage the findings to develop STADe. A dedicated gas wellhead monitoring testbed was designed to emulate real-world scenarios and facilitate data collection for attack simulations and analysis. Additionally, this research identifies a novel field flooding attack capable of disrupting critical OG processes.

This research emphasizes the significance of network traffic periodicity and demonstrates the effectiveness of anomaly detection models that leverage this characteristic.

Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
APM	Asset Performance Management
CPF	Central Processing Facility
C&C	Command-and-Control
CMMS	Computerised Maintenance Management Systems
CPS	Cyber-Physical Systems
DL	Deep Learning
DPI	Deep Packet Inspection
DoS	Denial of Service
DCS	Distributed Control Systems
DDoS	Distributed Denial of Service
DGA	Domain Generation Algorithms
EWS	Early Warning System
EPU	Electrical Power Unit
ESD	Emergency Shutdown Systems
E&P	Exploration and Production
FDR	False Discovery Rate
FN	False Negative

FP	False Positive
FPR	False Positive Rate
F&G	Fire and Gas Systems
ACLs	Firewalls and Access Control Lists
FPSO	Floating, Production, Storage, and Offloading
FC	Function Code
HIPPS	High Integrity Pressure Protection System
HMI	Human Machine Interface
HPU	Hydraulic Power Unit
ICS	Industrial Control Systems
ICPS	Industrial Cyber-Physical Systems
IIoT	Industrial Internet of Things
IT	Information technology
IOCs	International Oil Companies
IDS	Intrusion Detection Systems
KNN	K-Nearest Neighbour
LTI	Linear Time Variant
LNG	Liquefied natural Gas
LOF	Local Outlier Factor
LSTM	Long Short-term Memory
ML	Machine Learning
MitM	Man-in-the-Middle
MCS	Master Control Station
MAWP	Maximum Allowable Working Pressure

MBAP	Modbus Application header
OG/O&G	Oil and Gas
OPEX	Operating Costs
OT	Operational Technology
PCA	Principal Component Analysis
PLC	Programmable Logic Controller
PDU	Protocol data Unit
RTU	Remote Terminal Unit
SIS	Safety Instrumented Systems
STADe	Sliding Time-window Anomaly Detection
SCM	Subsea Control Module
SEM	Subsea Electronic Module
SCADA	Supervisory Control and Data Acquisition
TN	True Negative
TP	True Positive
VPN	Virtual Private Network

List of Publications

- A. S. Mohammed, P. Reinecke, P. Burnap, O. Rana, E. Anthi, Cybersecurity challenges in the offshore oil and gas industry: An industrial cyber-physical systems (ICPS) perspective, ACM Trans. Cyber-Phys. Syst. URL https://doi.org/10.1145/3548691

- Mohammed, A.S., Saxena, N. and Rana, O., 2022, May. Wheels on the modbus-attacking ModbusTCP communications. In Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (pp. 288-289)

- Mohammed, A. S., Anthi, E., Rana, O., Saxena, N., & Burnap, P. (2022). Detection and Mitigation of Field Flooding Attacks on Oil and Gas Critical Infrastructure Communication. Computers & Security, 103007. https://doi.org/10.1016/j.cose.2022.103007

- Mohammed, A. S., Anthi, E., Rana, O., & Burnap, P. STADe: An Unsupervised Time-Windows Method of Detecting Anomalies in Oil and Gas Industrial Cyber-Physical Systems (ICPS) Networks. *Under review.*

Dedication

To my loving wife and our three sons who endured this journey with me.

We did this!

Acknowledgements

I would like to take this opportunity to express my profound gratitude to the individuals who have played an instrumental role in shaping the course of my doctoral journey, making this research possible. Their unwavering support, expertise, and encouragement have been invaluable.

First and foremost, I extend my heartfelt appreciation to my primary supervisor, Professor Omer Rana. Professor Rana's rare insight, patience, and unique ability to guide me through the most challenging times have been a source of immense inspiration. His penchant for fresh ideas and the freedom he granted me to pursue my research allowed me to explore uncharted territories and discover new dimensions of my work. I am truly grateful for his mentorship and guidance, which have been the driving force behind this thesis.

I am also indebted to my additional supervisors, Professor Pete Burnap and Dr Eirini Anthi. Professor Burnap's unwavering commitment to excellence and candid feedback were pivotal in refining my research. His support in building the testbed for this project was indispensable and greatly appreciated. Furthermore, Dr Anthi's support was invaluable when it came to drafting and publishing our research papers. Her keen insights and dedication to the research process have enriched the quality and impact of our work. Of special mention is Dr Philipp Reinecke, my first primary supervisor, whose initial input at the start of this work proved pivotal in setting the pace. I remain grateful.

Special gratitude goes to my family: firstly, to my wife, Ruqayyah, for her sacrificed time and effort without which, I would never have been able to do this. I would also like to thank my parents for their continuous guidance and patience whenever I have erred. Also, to my sister, Sawdah, words aren't enough. Thank you. Special thanks also to my wider family for their encouragement.

I must extend my gratitude to all my amazing fellow researchers and friends who have supported me throughout this journey. Particularly my big brother Dr Ayo (the wise sage) whose advice always helped me recover from setbacks and disappointments. To my other PhD compatriots - Vasilis, Muzun and Fatimah, Mohammed Asiri, Naeima, Dalia, Areej, and Hakan - your encouragement, understanding, and numerous conversations have been essential in enabling me to pursue this research to its fullest potential. In addition, I would like to thank Dr Mike Lakoju for his encouragement. Special thanks also to David Sivell - who physically built the testbed; and to Jenny and Vijay who both helped with the testbed documentation.

Finally, I would like to express my appreciation to the Petroleum Technology Development Fund, Nigeria (PTDF) that provided financial support for this research. Their generous funding made this project possible.

This thesis is a testament to the collaborative efforts of many, and I am humbled and thankful for the guidance and support that have shaped my academic journey.

Chapter 1

Introduction

Despite global efforts to implement green energy sources, the demand for crude oil is expected to remain high for decades [1, 2]. Hence, it is crucial to protect the oil and gas (OG) industry from cyber threats [3]. While no business is immune, critical industries like O&G are increasingly vulnerable to cyber-attacks, especially in the operational domain [3]. This is because Operational Technology (OT) and Industrial Control Systems (ICS) that were once physically isolated from external networks [4] and used proprietary hardware, software, and communication protocols [5], are now connected to multiple industrial technologies, and integrated with Information Technology (IT) capabilities [6, 7, 8]. Consequently, ICS are at higher risk of cyber threats than before [9].

The future of oil and gas production involves accessing complex reserves, often located in deep and ultra-deep waters [10] in the form of offshore platforms. The more complex these reserves are, the higher the costs of production. Hence, O&G companies are having to devise cost-efficient solutions as a way around this. This has also driven oil and gas companies towards integration of IT and OT, creating Industrial Cyber-Physical Systems (ICPS) [11, 12] which has further exposed ICS communication protocols to cyber vulnerabilities like data exfiltration and malware injection attacks. These offshore platforms are sometimes unmanned facilities that require 100% remote monitoring which is dependent on digital networking communications. Since OG installations contain valuable and vulnerable operational data, they are a potential target for cyber-attacks that could compromise process safety, endanger the lives of personnel, damage the environment, and destroy the marine ecosystem [11]. A successful cyber-attack on any unmanned offshore platform, for example, would cause significant damage mainly because there is no option for a quick response due to the remote location.

1.1 Problem Statement

Due to the convergence of IT and OT, and the increase in security problems [13] a lot of security researchers have turned their focus towards industrial environments because of the importance of critical infrastructure. A number of studies have focused on various industrial sectors like manufacturing, food and chemicals, power grids, smart cities, and so on. However, very little has been dedicated to the oil and gas industry despite its pivotal role in the global economy.

Certain aspects of oil and gas operations have introduced unique vulnerabilities in the sector. The complexity, scale, and geographic distribution of offshore production platforms, for example, make it difficult to monitor and secure all systems effectively. One instance of this is subsea operational technology which is usually located in areas with limited access (e.g. middle of the ocean), reducing the ability to monitor, update, and secure them against potential threats. Additionally, geographical distribution across multiple countries and regulatory jurisdictions can lead to a lack of clarity of security arrangements, increasing the risk of cybersecurity incidents. These vulnerabilities, alongside others, will have to be identified and understood if we are to successfully protect this sector from threat actors.

Furthermore, because cyber-attacks on these industrial environments can be initiated from external sources (i.e. outside the network) and also within the local network, traditional security tools (e.g. firewalls) on their own may not be sufficient enough to detect internal threats. This makes it important to develop methods to monitor and analyse the behaviour of internal network communications from an industrial context. It is not unusual for attacks to go undetected in networked environments for several months prior to discovery and remediation. In an industrial environment, such a scenario could have catastrophic consequences as that would enable adversaries enough time to exfiltrate critical data that could be used to craft more sophisticated attacks.

1.2 Industrial Network Cybersecurity: Threat Vectors and Detection Methods

1.2.1 Threat vectors

ICPS is typically composed of three control components (i) Programmable Logic Controllers (PLC), (ii) Supervisory Control and Data Acquisition (SCADA), and (iii) Distributed Control Systems (DCS). However, one dominant component throughout all industrial control systems is the communication network, which connects all equipment and devices by electrical interfaces and communication protocols to ensure all systems communicate efficiently [14]. This makes the industrial communications network a very attractive target for threat actors to explore and represents an active pathway to compromising critical operations. To minimise this threat, the continuous development of efficient detection methods is critical as a first step. When a network is breached, how quickly the breach is detected is a critical factor in limiting the potential damage caused by the threat actors. As attackers are constantly devising new, complex tools to breach networks, there is a need for researchers to keep developing more efficient Intrusion Detection Systems (IDS) to cope with this rising threat.

One of the popular methods for attackers to gain entry into an industrial network is by exploiting the communications protocol. The Modbus protocol and its variants are the most widely used communications protocols in the oil and gas (OG) industry, especially for pipeline operations [15] and for monitoring remote offshore operations. The protocol was extended to allow control messages to be transported over TCP [16], creating the ModbusTCP variant. This hastened the wide adoption by the OG industry as communication could be integrated seamlessly within existing systems. Similar to other industrial protocols like DNP3 and OPC DA, the ModbusTCP protocol is insecure, lacking authentication or encryption, which makes it susceptible to cyber attacks (e.g. Man-in-the-Middle, Denial of Service, command injection, etc). The nature of OG operations, especially offshore production, requires remote monitoring of the production of highly volatile hydrocarbons from subsea to the surface. This requirement, together with the ease of deployment of

ModbusTCP to transmit sensor readings and actuator states has increased the widespread use of the protocol in the OG industry, and as a result, increased the attack surface of the Operational Technology (OT) being deployed.

1.2.2 Detection Methods

The popularity of intrusion detection systems is a result of the increase in network breaches by threat actors. An IDS serves a primary *'early warning system'* to system administrators informing them of suspicious activity on the network. There are two principal categories of IDSs - misuse/signature-based/supervised and anomaly detection/unsupervised [17, 18]. Machine learning (ML) is also commonly used in IDSs to distinguish between normal and malicious traffic [17].

Misuse-based IDSs use the information on known attacks to create rules – or in the case of supervised ML-based IDSs, utilise patterns of known attacks – to classify an event as either normal or benign [19]. Although this method can be highly accurate and efficient, it does have some weaknesses because of its inability to detect novel attacks (zero-day attacks) [20] and the difficulty of collecting labelled anomalous data for training and tuning the model [21]. However, when labelled attack data is available, misuse-based IDSs remain a very powerful and efficient tool to detect cyber attacks. This is evidenced by most commercial systems that only deploy it in their security tools [17].

Anomaly detection or unsupervised IDSs, on the other hand, work on the principle of learning the normal behaviour or pattern of a system and classifies traffic as an anomaly by identifying a deviation from normal patterns [22, 23]. It does not require labelled data and is able to detect zero-day attacks. One of its main disadvantages, however, is the generation of high false positives [24] and false alarms which can potentially overwhelm security analysts.

1.3 Research Contributions

The broad contribution of this thesis is to demonstrate the effectiveness of supervised and unsupervised methods of detecting cyber-attacks in an oil and gas industrial network environment. To do this, this thesis aims to answer the following research questions which are raised to address the problems identified earlier:

RQ1 What common vulnerabilities, popular protocols, and attack patterns are prevalent in the oil and gas industry?

RQ2 To what extent can vulnerabilities in prevalent communication protocols employed within the oil and gas industry be exploited further?

RQ3 How well do supervised machine learning methods fare in identifying cyber-attacks against a typical oil and gas OT setup?

RQ4 How efficiently can unsupervised machine learning methods be utilized to detect anomalies in industrial cyber-physical systems?

RQ5 How can the inherent periodicity within industrial networks be effectively leveraged to enhance anomaly detection?

RQ6 Is the investigated unsupervised anomaly detection method adaptable across different industrial networks?

In answering these questions, the following contributions are made:

C1 This thesis contributes an extensive survey on the cybersecurity challenges in the offshore oil and gas industry. The O&G production process and its vulnerabilities to cyber-attacks are described as well as the limitations in available datasets and testbeds for security research on OT infrastructure.

C2 Design and installation of a wellhead monitoring testbed to emulate the oil and gas production process and aid cybersecurity research in the OG industry.

C3 This research identifies a novel “Field Flooding” attack on the ModbusTCP protocol which can lead to a severe Denial of Service (DoS) attack.

C4 This research evaluates an Intrusion Detection System to effectively detect the Field

Flooding attack on industrial control networks using a supervised machine learning approach.

C5 An investigation into how unsupervised machine learning algorithms can be utilised for anomaly detection in industrial cyber-physical systems.

C6 This research contributes a catalogue of labelled industrial network datasets in csv format including the original pcap files containing benign and attack data. The attacks carried out are field flooding attacks, SYN flooding attacks, and Man-in-the-Middle attacks.

C7 This research contributes a novel methodology of unsupervised Time-Series Method of Detecting Anomalies in Industrial Cyber-Physical Systems (ICPS) Networks.

C8 This research determines the adaptability of the STADe methodology described in Chapter 6 to different industrial cyber-physical networks.

1.4 Thesis Structure

The outline for the remainder of this thesis is as follows:

- **Chapter 2 - Background:** This chapter surveys the challenges with securing offshore oil and gas assets from cyber-attacks from an operational perspective and forms the main motivation for this research. Further, the oil and gas production process and its vulnerabilities to cyber-attacks are described. Furthermore, this chapter highlights the limitation of available industrial datasets and testbeds for security research on operational technology infrastructure. This chapter presents contribution **C1**.

- **Chapter 3 - Methdology:** In this chapter, the research methodology and general approach was described which highlighted the focus on developing and evaluating anomaly detection methods tailored for industrial network environments such as oil and gas critical infrastructure.

- **Chapter 4 - Field Flooding Attacks – Detection using supervised machine learning:** In this chapter, a novel field flooding attack which is capable of causing a denial of service on devices using the ModbusTCP protocol is described. In addition, the chapter also describes the evaluation of supervised machine learning algorithms to be utilised in an intrusion detection system that is capable of detecting the novel attack. This chapter presents contributions **C2**, **C3** and **C4**.

- **Chapter 5 - Unsupervised Machine Learning Methods of Detecting Anomalies:** Building on the previous chapter, this chapter investigates unsupervised machine learning algorithms and their effectiveness in detecting anomalies in industrial networks. Further, network datasets are created with popular attacks known to target oil and gas systems which are described. This chapter presents contributions **C5** and **C6**.

- **Chapter 6 - STADe: An Unsupervised Time-Windows Method of Detecting Anomalies:** In this chapter we develop a novel methodology STADe to detect anomalies in industrial networks by defining the periodicity in a given industrial network. In addition, the chapter also discusses techniques to visually represent this periodicity as a pattern in 3-dimensional space. This chapter presents contribution **C7**.

- **Chapter 7 - Testing Robustness and Adaptability of STADe Performance Results on Public Datasets:** This chapter assesses the adaptability of the STADe methodology developed in Chapter 6 in different industrial network environments and evaluates its effectiveness in detecting anomalies in such networks. This chapter presents contribution **C8**.

- **Chapter 8 - Conclusions:** This chapter summarises the research carried out in this thesis and highlights possible future work.

Chapter 2

Background: Cybersecurity Challenges in the Offshore Oil and Gas Industry - An Industrial Cyber-Physical System (ICPS) Perspective

Parts of this chapter have been published in the paper ”Cybersecurity challenges in the offshore oil and gas industry: An industrial cyber-physical systems (ICPS) perspective”, ACM Trans. Cyber-Phys. Syst. URL https://doi.org/10.1145/3548691

2.1 Introduction

This chapter introduces oil and gas operations and describes the cybersecurity challenges that have arisen from operational vulnerabilities. More importantly, this chapter contains our first contribution:

C1 *This thesis contributes an extensive survey on the cybersecurity challenges in the offshore oil and gas industry. The O&G production process and its vulnerabilities to cyber-attacks are described as well as the limitations in available datasets and testbeds for security research on OT infrastructure.*

The purpose of this chapter is twofold: first, to examine the nature of oil and gas operations and analyse the potential vulnerabilities in an end-to-end sub-system (i.e. subsea control system). Second, to provide insight into the history of cyber attacks on oil and gas assets, challenges in securing these assets, and existing detection techniques employed in previous research. Finally, this chapter suggests avenues for further investigation which highlights the importance and motivation for the rest of this thesis.

The survey is organised as follows: First, the need for remote monitoring in oil and gas (OG) facilities is discussed as well as the comparison of the OG industry and other critical infrastructure industries with respect to cyber vulnerabilities, followed by an overview of the upstream sector of the O&G industry and a description of the O&G production process. From this, common vulnerabilities, attack vectors in the sector, and a case study of a subsea control system architecture are discussed. An explanation around challenges and analyses on why upstream O&G assets are difficult to secure is elaborated on, as well as the state of securing O&G assets including datasets available for security research.

2.2 The need for remote monitoring: How vulnerable is the OG industry to cyber attacks?

The integration of OT and IT has been aided by the rapid development of embedded systems, sensors, and networks, which in turn has given rise to Cyber-Physical Systems (CPS) [25]. An Industrial Cyber-Physical System (ICPS) refers to CPS that is specifically designed for industrial applications [25]. This has opened the door to significant efficiency gains in the oil and gas industry[26] and is particularly the case in the offshore sector, where there is a pressing need to reduce costs and maximize equipment availability [26]. While it allows engineers to monitor and control assets remotely [6], [27], [28], this also exposes ICS communication protocols to vulnerabilities – such as data exfiltration and malware injection attacks. These vulnerabilities could cause significant losses to a company and potentially compromise process safety; endangering lives of personnel including damage to the environment. The migration to IT has also led to the standardisation of new SCADA communication protocols such as Modbus-TCP, Distributed Network Protocol (DNP3), IEC-60870–5-104 and the Inter-Control Center Protocol (ICCP, IEC60870–6) [27]. The first three were designed for automation and control, and the last was designed to interconnect SCADA systems [27]. Figure 2.1 shows an example of typical components that make up an ICPS in an offshore O&G platform and highlights common vulnerabilities.

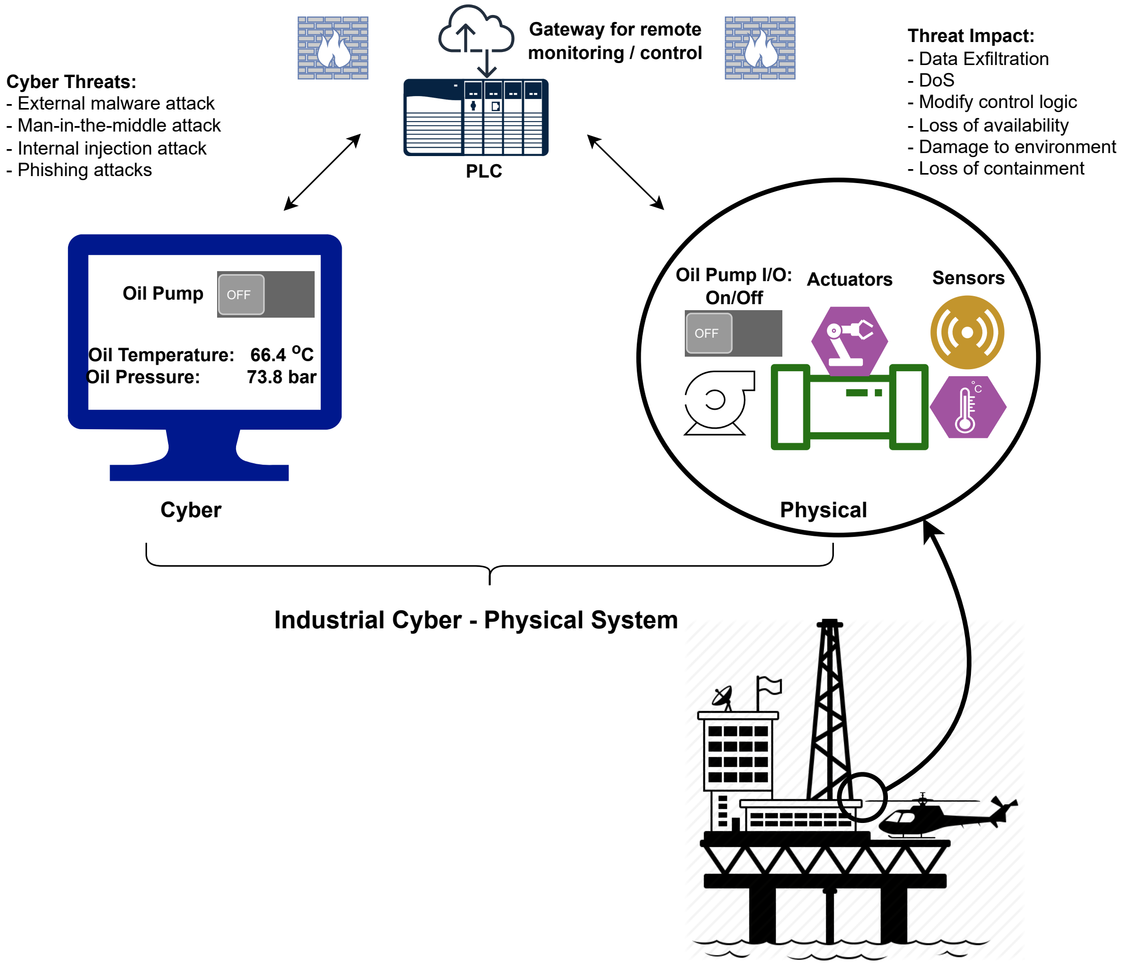


Figure 2.1: Example of a Cyber-Physical System in oil and gas highlighting common vulnerabilities

O&G production and processing facilities rely heavily on ICPS [29]. Control equipment such as Programmable Logic Controllers (PLCs) and Distributed Control Systems (DCS) are widely used, along with Human Machine Interfaces (HMIs) and Remote Terminal Units (RTUs) [29]. Using Industrial Internet of Things (IIoT) technology, the interconnection of these intelligent industrial devices with control and management platforms, collectively, improve the operational efficiency and productivity of industrial systems [30]. One of the more common use case of ICPS in the O&G industry that depend on these control equipment is Asset Performance Management (APM) – a data-driven approach to asset management [26]. APM solutions are often linked to Computerized Maintenance Management Systems (CMMS) which also results in their deployment on-premise [26]. As the move toward minimally manned facilities continues, having remote visibility into operations becomes increasingly important [26]. Currently, the need to transfer production data to information systems, which also includes the needs for remote maintenance [29] has helped in broadening the attack surface across the O&G industry.

Due to their remote location in deep waters and the need for real time monitoring and control, offshore O&G assets potentially have a larger attack surface compared to other sub-sectors of the industry, which makes them attractive to threat actors. This is critical because offshore production accounts for a significant proportion (about 30% [31]) of global O&G production. There also seems to be a passive shift in focus towards offshore production in some oil producing countries. In Nigeria, for example, some International Oil Companies (IOCs) are divesting their onshore producing assets to focus more on deep offshore production [32], [33]. Additionally, Equinor (a company focusing on Petroleum refining) has a large portfolio of offshore assets in the US Gulf of Mexico, and has agreed to divest its onshore assets in the Bakken Field [34]. These trends are consistent across continents, and indicate that the offshore O&G sub-sector is likely to retain or increase its share of global O&G production.

Successful cyberattacks threaten the competitiveness of the global O&G industry, and the cost of future breaches will be much higher, whether to corporate assets, public infrastructure and safety, or the broader economy through energy prices [35]. Breaches can lead to lost production, raised health, safety and environmental risk, costly damages claims, breach of insurance conditions, negative reputational impacts, and loss of licence to operate. Therefore, cybersecurity needs to be a consideration throughout the life-cycle of any project, especially across digital transition activity of the O&G sector [36].

The reported percentages of acknowledged cyberattacks indicate the high threat for offshore O&G assets [8]. In addition to the attacks identified above, a cyberattack on an O&G OT environment can have serious results beyond just financial losses including en-

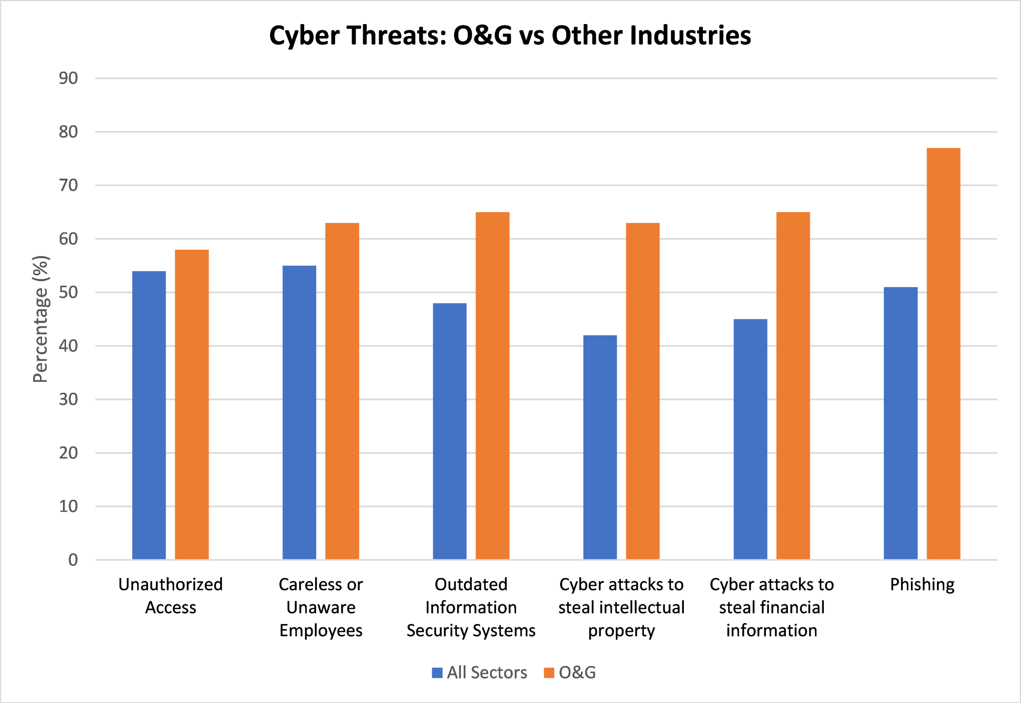


Figure 2.2: Cyber threats faced by O&G sector as compared to other industrial sectors. Source: EY Global Information Security Survey 2016-17 [35]

vironmental damage such as direct manipulation of machinery [6], changes to inclination of entire oil rigs [37] or pressurisation of pipelines [38], [39].

2.2.1 O&G vs Other Critical Infrastructure Industries

In 2017, EY carried out a 2016-17 Global Information Security Survey shown in Figure 2.2 where selected companies were asked which threats and vulnerabilities have most increased their risk exposure over the last 12 months. In every single metric recorded, the O&G companies had a higher cyberattack incidence frequency compared to other critical infrastructure industries. More specifically, in the United States, the Department of Homeland Security responded to more than 350 cyberattack incidents at US energy companies between 2011–2015, and identified nearly 900 security vulnerabilities within those energy companies – higher than any other industry [29] [40].

Another study that examined the state of cybersecurity in the United States O&G industry was carried out by The Ponemon Institute [41] in 2017, where 377 individuals who were responsible for securing or overseeing cyber risk in the OT environment were surveyed. It was discovered that only 41% continuously monitor all infrastructure to prioritize threats and attacks. An average of 46% of all cyberattacks in the OT environment go undetected, suggesting the need for investments in technologies that detect cyber threats to O&G operations [41].

The vulnerability of this sector was most evident in May 2021, when one of the largest pipelines in the US – the Colonial Pipeline – which carries refined gasoline and jet fuel from Texas along the East Coast of the US to New York, was forced to shut down its 5,500 miles of pipelines for six days due to a cyberattack [42]. Reports indicate that this was a ransomware attack that targeted the IT system, yet its repercussions were felt in OT operations as headline news reported panic, social disruption and a crippling lack of fuel delivery [43].

Furthermore, O&G companies frequently withhold specific financial details regarding losses from cyber-attacks. This lack of transparency could be driven by a combination of reputation management, market stability, regulatory concerns, and security risks. As a result, we can only estimate the full impact of these incidents based on tangible operational costs. For example, after the Colonial Pipeline incident, the company disclosed a ransom payment of $4.4 million but did not detail the broader economic impacts of the operational shutdown[44]. However, although no detailed analysis was made public, the overall economic impact of the attack is estimated to be in the range of hundreds of millions of dollars, factoring in lost productivity, increased fuel prices, and the cost of the company's operational disruptions and response measures [44].

These events highlight the increasing severity of the problem, and also presents an open question in oil and gas cybersecurity:

RQ1: What common vulnerabilities, popular protocols, and attack patterns are prevalent in the oil and gas industry?

To answer this, it is important to survey previous studies and highlight the gaps that form the broader motivation for this thesis.

2.2.2 Related Work

Although significant research has been conducted on SCADA systems in general (e.g. [27], [45]), only a handful focus on investigating the security status of O&G. These studies investigate the problem as part of a multi-industry study without specific context to O&G operations. Very few publications survey cybersecurity topics specifically for the O&G sector [6], although some [56], [57], [58] survey cybersecurity incidents related to the O&G industry [6]. The relevant existing (key) literature has been summarised in Table 2.1. Stergiopoulos et al. [6] carried out a study that focused on analysing and understanding past attacks and vulnerabilities in the O&G sector based on documented cybersecurity incidents and developed a vulnerability taxonomy for ICPS specifically for the O&G sector. The concepts presented in the study are broadly applicable to the 3 sub-sectors in O&G: upstream, downstream and midstream on a generic level. However, Stergiopoulos et al. [6] do not consider the following aspects:

- potential threats and vulnerabilities specific to upstream operational systems, and
- potential mitigation strategies specific to these threats to the system.

This chapter extends their study by analysing the key components in a subsea control system – which are usually designed to different standards from onshore platforms due to the extreme conditions that exist in deep waters. To the best of our knowledge, this study is the first to analyse the vulnerabilities of a subsea control system to cyberattacks. The research to date has focused generally on ICS security, which is broadly applicable to most sectors with only a few analysing real cybersecurity incidents that have taken place in the O&G sector. Studies on O&G ICPS security have lacked domain knowledge of a complete end-to-end process system, while highlighting vulnerabilities. This is important because showing vulnerabilities of specific existing engineering designs used in the field could lead to more resilient systems. From a literature review perspective, it is evident that the subject of cybersecurity for O&G assets is not widely studied [8]. Reports also

Author / Reference	Summary	Multiple Sectors	O&G Specific	Threats to Upstream Operations	Proposed Mitigation to Threats
Alcaraz et al. [27]	Architectural components of critical infrastructure components and their vulnerabilities	✓			
Kim et al. [45]	Survey of CPS research in hybrid systems, security & real-time computing. Outlined potential for CPS in several applications	✓			
Krotofil et al. [46]	ICS security research including controls to mitigate vulnerability of common ICS protocols.	✓			
Mo et al. [47]	Survey of information security approaches for cyber-physical systems	✓			
Stergiopoulos et al. [6]	Attack taxonomy and catalogue of cyberattacks on O&G assets		✓		
McLaughlin et al. [48]	Overview of ICS security including key principles of ICS operations	✓			
Sadeghi et al. [49]	Security and privacy issues for IIoT with proposed mitigations	✓			
Stellios et al. [28]	IIoT threat landscape analysis with representative attacks against IoT	✓			
Khan et al. [50]	IoT architecture specifically for the O&G industry to aid functional and business requirements		✓		
Sayegh et al. [51]	A testbed used to detect vulnerabilties in SCADA protocols	✓			
Nazir et al. [52]	Survey of tools and techniques to discover SCADA system vulnerabilities	✓			
Bhamare et al. [53]	Explored major publications from industry and academia and addressed applicability of machine learning techniques for ICS cybersecurity	✓			
Miller et al. [54]	Cybersecurity incidents on critical infrastructure and SCADA systems and a taxonomy to classify future SCADA security incidents	✓			
Giraldo et al. [55]	Classification of CPS domains, security level implementation and computational strategies	✓			
This survey	Survey identifying specific cyber threats and vulnerabilities to upstream O&G assets and mitigation strategies		✓	✓	✓

Table 2.1: Summary of Related Work

indicate that the industry's *cyber maturity* is relatively low, and O&G boards show very little understanding of cybersecurity requirements [59], [6].

2.2.3 Motivation

Out of 42 recorded cyber security incidents [6] affecting the O&G industry in the past decade, the upstream sector had the highest number of incidents. This gives an indication of a higher vulnerability in this sector compared to other O&G sub-sectors. Moreover, because the upstream sector is the first stage of 3 highly inter-connected sectors of the O&G industry (which will be described briefly in Section 2.3), any disruption will likely cascade down the value chain and have an impact on the other sectors. For these reasons, this chapter will be focused on the upstream O&G sector – addressing two questions: (i) what unique challenges make the industry more vulnerable to attacks compared to others? and (ii) why the available datasets for cybersecurity research are largely not representative of the O&G industry processes. To the best of our knowledge, there has been no survey carried out specifically for the offshore environment of the O&G industry, identifying inherent vulnerabilities in an end-to-end subsea system. This chapter describes the O&G production process and its vulnerabilities, presents a timeline of documented cyberattacks on O&G upstream assets, analyses a subsea control system, highlighting the vulnerabilities of the system to cyberattacks, identifying mitigation strategies against such vulnerabilities, and discusses limitations in available datasets for security research on OT infrastructure.

2.3 Overview of the Oil and Gas Industry

The O&G industry comprises of three sub-sectors: upstream, downstream, and midstream infrastructures [6]. These sectors are quite diverse in their roles within the value chain. The ***upstream*** sector deals with exploration, drilling and production [6], i.e. all activities involving the search for oil/gas, the recovery process and production from reservoirs at very high pressures and temperatures. It comprises of offshore and onshore operations. The ***downstream*** sector focuses on distributing assets to consumers [6] and handles the refining of the natural gas or crude oil produced and its storage facilities (oil refineries, Liquefied Natural Gas plants, gas stations, petrochemical plants, etc.) while the ***midstream*** connects the upstream activities to the downstream activities [6], i.e. transportation – pipelines, crude oil tankers, trucks; and marketing activities. These three sectors are interconnected and interact through a complex web of activities which are streamlined to ensure a timely and safe delivery of petroleum products to end consumers. These sectors are highlighted in Figure 2.3. In the next sub-sections we will describe the life cycle of an O&G upstream asset and the associated production process.

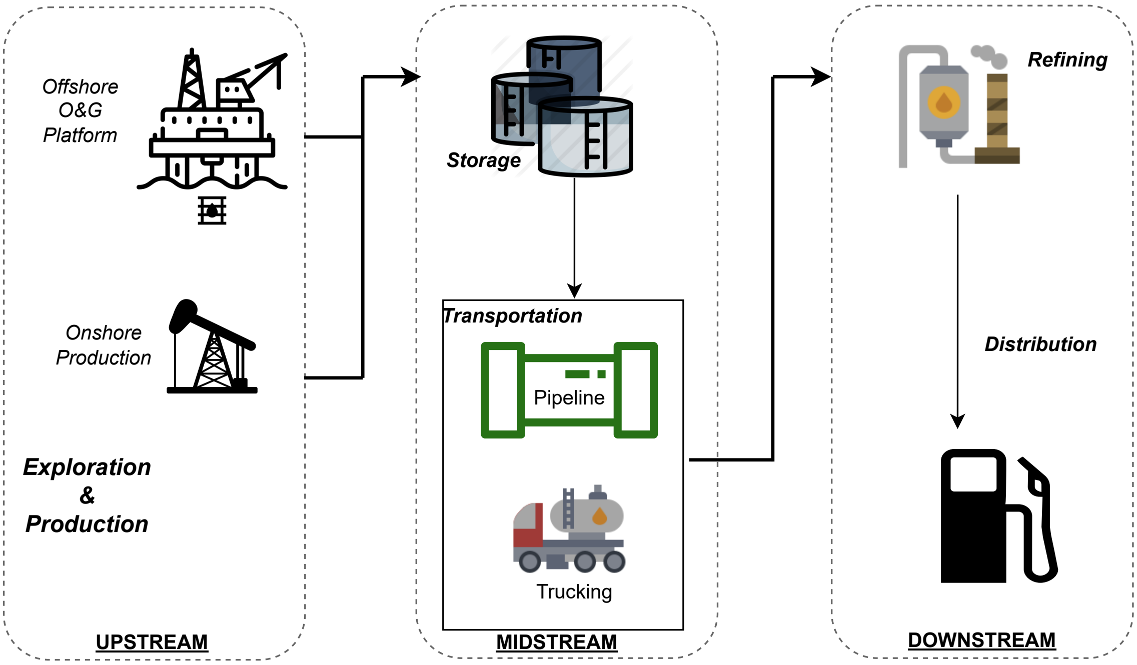


Figure 2.3: Oil and Gas Value Chain

2.3.1 Life Cycle of the Upstream Oil and Gas Industry

Upstream activities include exploration, drilling, and production and are typically referred to as E&P (Exploration & production). More specifically, the upstream life cycle is split into five phases which cover the ’cradle to grave’ activities ranging from how hydrocarbons are discovered to reservoir depletion, and decommissioning (returning the environment to its pre-E&P state). The activities that take place in each phase and their average timelines are summarised in Table 2.2.

Phase	Timing	Activities
1. Exploration	1-5 years	Exploration for potentially viable oil and gas sources through geological surveys. Operations are terminated if no viable sources found. [60].
2. Appraisal	4-5 years	Potential sites containing viable oil/gas sources [60].
3. Development	4-10 years	Limited infrastructure and site development will already be in place as part of the exploratory and initial drilling phase, but during the field development phase activity will dramatically increase and first oil/-gas will be produced towards the end of this phase [60].
4. Production	20-50 years	Oil/gas reserves are extracted and transported for processing and distribution [60].
5. Decommissioning	2-10 years	The platform may be removed and the seafloor returned to its pre-lease condition [61]. Once it is no longer cost-effective to extract remaining reserves, the site is decommissioned, with operating companies being responsible for returning the site to as close to original state as possible [60].

Table 2.2: Upstream life cycle describing activities carried out during each phase

2.3.2 Upstream O&G Production and Processing

O&G production is the process of extracting reservoir fluids (hydrocarbons) and separating the mixture of oil, gas and water at the surface. The main activities are gathering (from wellheads to separators), separation, gas compression (to prepare for storage and transport), temporary oil storage, waste water disposal and metering (calculation of quantity before export) [62]. From the wellheads, reservoir fluids are fed into production and test manifolds. Next stage is the separation process, where horizontal gravity separators are usually used [63] in most facilities. The fluids are separated based on their densities (water is heavier than oil while gas is the lightest). In the separator, the pressure is often reduced in several stages, from high pressure to low pressure, to allow controlled separation of volatile components [63]. The gas is dehydrated, compressed and used to power the plant in most cases while the rest is exported. The oil is also processed and stored in settling tanks ready for export while the produced water could be re-injected into the reservoir for pressure maintenance or disposed off safely. There are a number of variations to this process depending on the crude oil composition and the required end products, but this is the typical baseline setup for most O&G production facilities. This process is illustrated in Figure 2.4.

2.3.3 Offshore Operations

Offshore O&G operations are a subset of upstream operations. It is common for offshore O&G operators to have a service territory that spans a large geographic area [29]. A large O&G company operating offshore, for instance, generates, transmits and stores petabytes of sensitive and competitive field data; and operates and shares thousands of drilling and production control systems spread across geographies, fields, vendors, service providers and partners [64]. Most of the field data transmitted and stored are collected by sensors that are part of an industrial control system. ICPS sits at the heart of remote operations which enables the satellite platforms to be fully automated. For this reason, central operation centres may be constructed to control system flow and monitor system conditions [29], which is made possible by utilising ICPS for collection of data and control of critical system processes. A large offshore oilfield development project would typically have several types of platforms to effectively extract and export O&G resources from the deep oceans. These structures would be distributed around the field(s) (several kilometers apart) as satellite platforms. The reservoir fluids extracted would be transported via pipelines to a central processing facility (CPF) where they are processed, stored, then offloaded to export tankers. The extraction of crude oil from offshore facilities is made possible by subsea control systems, and in recent times subsea production systems. These are highly advanced equipment designed to operate under extreme pressures and temperatures found in deep waters.

Drilling Campaigns: Throughout the life of a field, there will be several drilling campaigns carried out. During the exploration phase, drilling is used to find commercial quantities of hydrocarbons. In the appraisal phase, drilling is used to confirm how large the reservoir is and its characteristics. In the development phase, drilling is more precise as this is where the initial production wells will be drilled. During production, there may be in-fill drilling to improve the efficiency of depleting the reservoir by adding more wells during the life of the field. Drilling rigs therefore move between fields globally to execute drilling campaigns. When a drilling rig arrives on site, it can be attached to a host platform for shared resources. This is usually taken into account when designing offshore structures. Drilling operations are usually carried out by oil service companies. They are different from the company who owns and operates the O&G asset. A common use case for ICPS during drilling operations is to enable leak detection, in which a remote multi-sensing technology [65] could be used. This helps in identifying potential leaks and aids quick response to limit the release of harmful hydrocarbons into the environment.

2.3.4 Remote Offshore O&G Production Operations

While most offshore platforms are still currently manned facilities, there is an emerging trend towards a shift in operating oil rigs completely remotely from land. Several recent studies and innovations supporting unmanned O&G production have also indicated this shift [66], [67], [68], [69], [70]. Some examples of such studies are the DNV unmanned floating LNG (Liquified Natural Gas) concept, Solitude and Aker Solutions' conceptual idea for an unmanned FPSO with annual maintenance campaigns [71]. Equipment is modularized and monitored from shore for routine maintenance and fault correction carried out by self-programming autonomous inspection and maintenance units [1]. Hence, the ability to operate an unmanned platform as part of a portfolio of offshore assets leads to reduced operating costs (OPEX) [72]. This is a huge factor influencing O&G companies operating offshore to invest in this technology, which also has the potential to increase potential cyber threats against such systems.

2.4 Common Vulnerabilities and Attack Vectors in the Upstream O&G Industry

In Section 2.3, the basic process flow in O&G production was described. There are inherent vulnerabilities that an attacker could exploit in the system. In this section, we will discuss the types of attacks that can compromise the system and present a case study of subsea control, communications, and its common vulnerabilities.

Process monitoring and remote control are two common activities that utilise ICPS and automation to optimise operations. These are generally applicable to:

1. **Monitoring (Sensors):** Temperature, pressure, chemical composition, leak detection, etc.

2. **Remote Control:** Valves/ actuators, pumps, hydraulic and pneumatic control systems, Safety Instrumented Systems (SIS), Emergency Shutdown Systems (ESD), Fire & Gas Systems (F&G), High Integrity Pressure Protection System (HIPPS), etc.

The following sub-section will examine the attacks that could compromise any of these operations.

2.4.1 Types of Attacks

- **Denial of Service (DoS):** One of the main safety features for process control are Emergency ShutDown systems (ESD) which are used to mitigate unsafe operating conditions. ESD systems in O&G platforms typically communicate using the Modbus protocol – an insecure communication protocol, as it lacks authentication and sends data without encryption. In a DoS attack, an attacker could take advantage of this by sniffing network traffic to understand the rate of communication and range of sensor readings, then crafting malicious packets similar to legitimate Modbus requests with the aim of flooding the network to render the ESD PLCs incapable of responding to unsafe process control requests. If an attacker, for example, were to carry out a DoS attack on the ESD of an unmanned offshore oil facility, a major catastrophic event could happen if there was a pressure build-up in the crude oil export lines. This kind of attack ensures that the onshore control centre loses its ability to shut down critical process to avert danger.

- **Oil Tank Level Spoofing Attack:** Processed oil that has been treated and separated from gas and water is stored in settling tanks ready for export. These tanks are fitted with level control sensors that transmit information to prevent tank overfills.

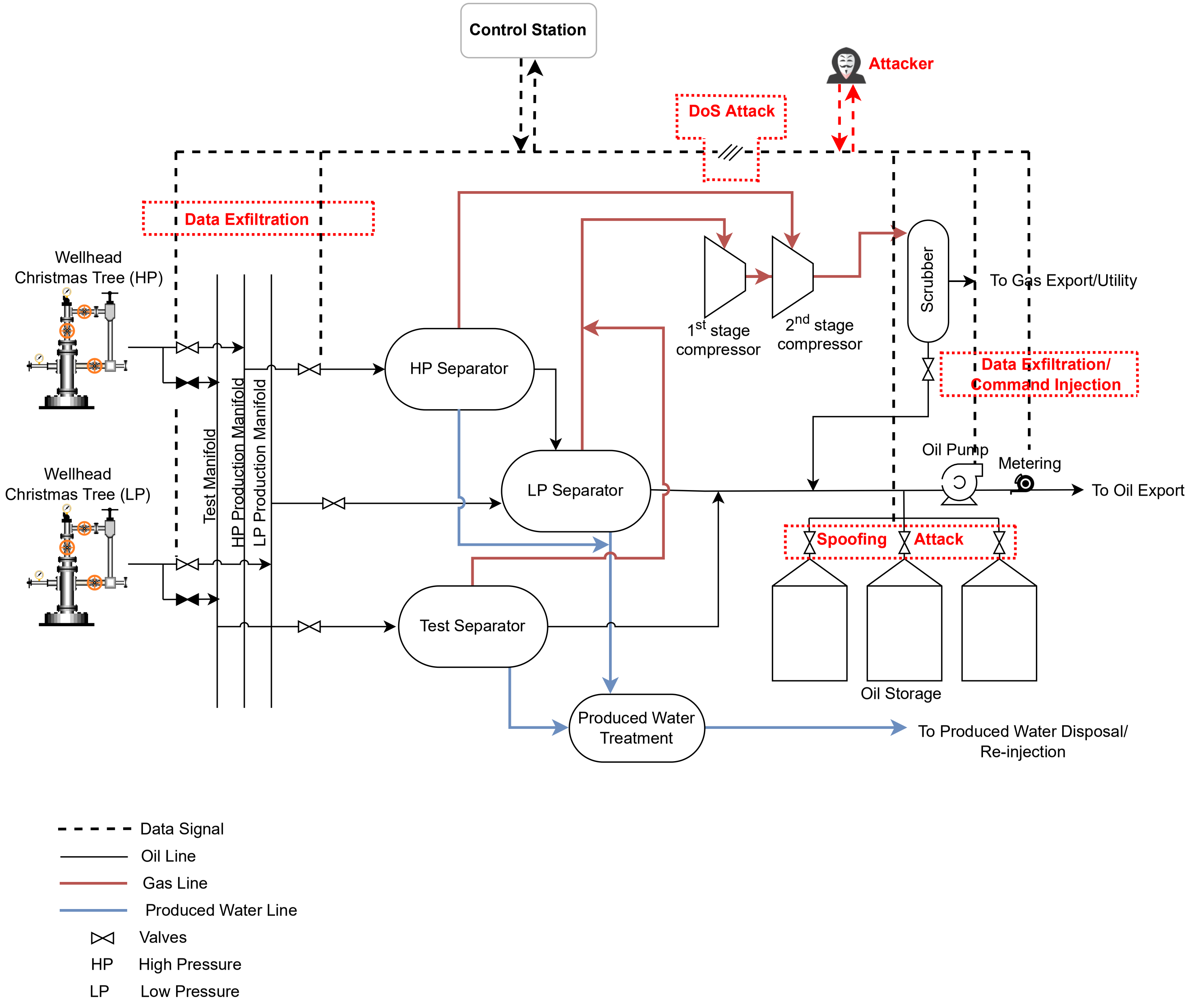


Figure 2.4: Upstream oil and gas production process showing potential cyberattacks

The main goal of this attack is to falsify sensor readings (e.g. Man-in-the-Middle attack) indicating that the tank level is lower than it actually is, which could lead to explosions due to a tank overfill as oil is a highly volatile product.

- **Wellhead Production Data Exfiltration:** By discretely deploying malicious software such as trojans on compromised workstations in the control station, an attacker could be privy to sensitive information like wellhead production data. There are various ways a threat actor could harvest sensitive company data using stealthy techniques. An example is with the use of Domain Generation Algorithms (DGA) in establishing communications between bots and their Command-and-Control (C&C) servers. Accessing metering data at custody transfer points also provides an attacker with sensitive information. This could allow threat actors to study hydrocarbon export volumes over time and provide them with enough information to prepare stealthy spoofing attacks that could cause loss of revenue to the company. Some companies have had their data discretely exfiltrated for years before it was found out.

- **Command Injection:** PLCs control numerous operations in the oil production process described in Section 2.3.2. An example is the oil export system which comprises of export pumps, flow computers, flow meters and actuators. If an attacker were to compromise an engineering workstation in the control centre, they could alter legitimate commands to cause the pump or actuators to perform inappropriately. In addition, PLCs are programmed to control the process to perform within safe operational parameters like maximum allowable pressure and flowrate. These set point limits, if tampered could lead to unsafe operational states. O&G being volatile hydrocarbons need very little instability to ignite and cause explosions.

- **Data Tampering:** Processed data could be tampered with by an attacker. An attacker could obfuscate the details of a wider attack by altering operation log and system control-related data [73], which would deceive defenders carrying out a post-attack forensic analysis. Data historians in offshore control stations that store operation log files could be targeted by this attack.

- **Choke Size Replay Attack:** This is a type of replay attack where the signed packets sent over the network could be captured and resent multiple times to the destination [74]. An example of a dangerous application is if an attacker were to intercept commands sent to increase or decrease the choke size of a well (to increase or decrease crude oil production rates). They could replay these commands to increase the choke size, masking as a legitimate command, which could damage the reservoir permanently.

Table 2.3 summarises potential attacks on upstream O&G processes showing attacks, attacker motives, vulnerable components and potential consequences including the impact of attack. These vulnerable points in the oil production process are also highlighted in Figure 2.4.

Based on the recorded security incidents [6] affecting upstream O&G assets, Figure 2.5 shows a pattern that indicates that these vulnerabilities are already being exploited, and that threat actors have this capability. The most frequent impact from these attacks were theft of operational information (8 incidents) and DoS (6 incidents). Using the CIA triad of Confidentiality, Integrity, and Availability, these can be represented as attacks on confidentiality and availability respectively. Unlike IT environments, Availability is considered the most critical aspect of cybersecurity in OT environments [75]. This makes DoS attacks in industrial environments such as oil and gas facilities highly disruptive with potentially fatal consequences. The impact of these types of attacks on a subsea control system is investigated in Section 2.4.2.

2.4.2 Subsea Control and Remote Monitoring: A Case Study

One of the critical processes in offshore operations is the subsea control system. Located hundreds of metres under deep waters, this system is essentially responsible for real time monitoring of production parameters to prevent unsafe conditions. We have focused on an offshore system because our literature analysis indicates that the upstream sub-sector

O&G Process	Attacker Motive	Potential Attack	Component	Consequence	Impact
Oil tank storage	Service disruption	Spoofing	Level sensors	Tank overfill, loss of containment	Explosion, loss of life, environmental damage
Hydrocarbon separation	Revenue loss	Data tampering	Pressure or Temperature sensors	Incomplete separation of gas from oil	low-quality product, loss of revenue
Oil delivery, export, piping	Service disruption	Command injection	PLC, pumps, actuators	Operations outside allowable limits	Potential damage to asset and environment, potential loss of lives
Emergency Shutdown	Damage to asset	DoS	Safety Instrumented System, PLC, and actuators	Operations outside allowable limits	Potential damage to asset and environment, potential loss of lives
Custody Transfer/ Metering	Revenue loss, theft of operational information	Data exfiltration	flow computers, meters, pressure/ temperature sensors	Incorrect calculation of hydrocarbon volumes, sensitive operational data leakage	Loss of revenue, reputational damage

Table 2.3: Attacks on some O&G upstream processes showing attacker motives and impact

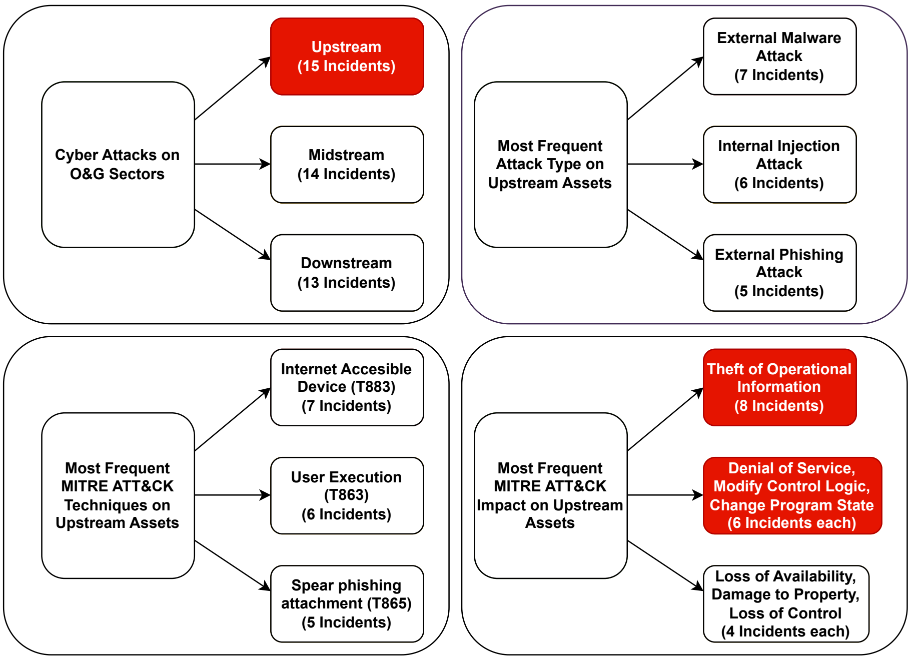

Cyber Attacks on O&G Sectors
Upstream (15 Incidents)
Midstream (14 Incidents)
Downstream (13 Incidents)
Most Frequent Attack Type on Upstream Assets
External Malware Attack (7 Incidents)
Internal Injection Attack (6 Incidents)
External Phishing Attack (5 Incidents)
Most Frequent MITRE ATT&CK Techniques on Upstream Assets
Internet Accesible Device (T883) (7 Incidents)
User Execution (T863) (6 Incidents)
Spear phishing attachment (T865) (5 Incidents)
Most Frequent MITRE ATT&CK Impact on Upstream Assets
Theft of Operational Information (8 Incidents)
Denial of Service, Modify Control Logic, Change Program State (6 Incidents each)
Loss of Availability, Damage to Property, Loss of Control (4 Incidents each)

Figure 2.5: Analysis of cyberattacks on upstream assets; adapted from [6]

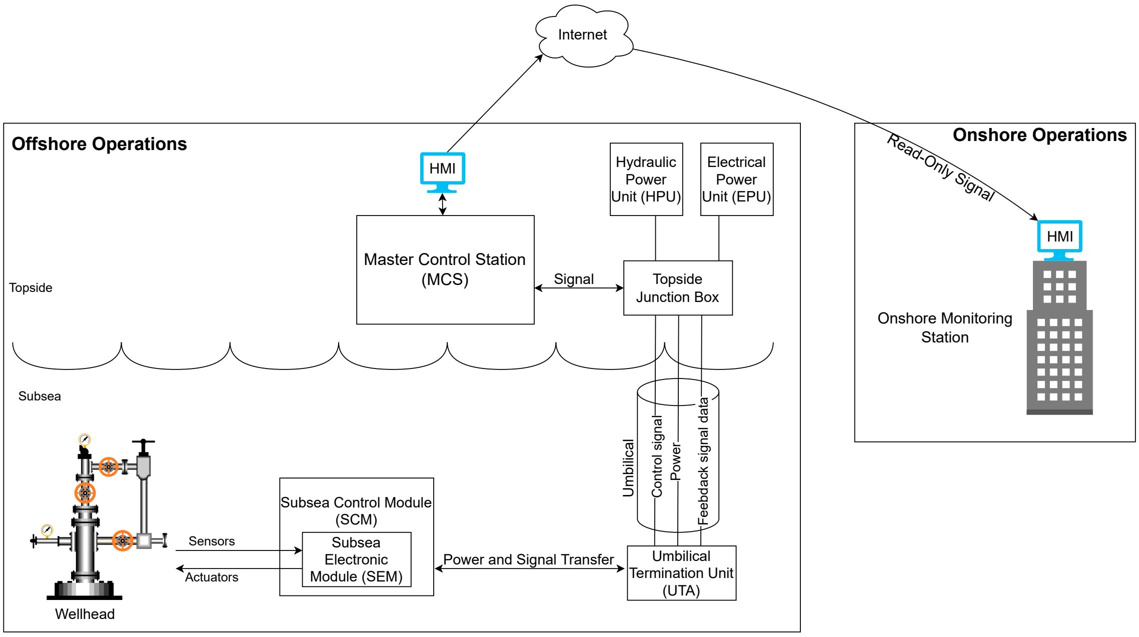

Internet
Offshore Operations
HMI
Hydraulic Power Unit (HPU)
Electrical Power Unit (EPU)
Master Control Station (MCS)
Signal
Topside Junction Box
Topside
Subsea
Umbilical
Control signal
Power
Feebdack signal data
Subsea Control Module (SCM)
Subsea Electronic Module (SEM)
Sensors
Actuators
Power and Signal Transfer
Umbilical Termination Unit (UTA)
Wellhead
Onshore Operations
Read-Only Signal
HMI
Onshore Monitoring Station

Figure 2.6: Example of an offshore subsea production monitoring system

is more vulnerable to attacks. Furthermore, an attack on a physical process in an offshore (possibly unmanned) asset will take a much longer time to respond to, compared to an onshore asset. An example of a typical setup is shown in Figure 2.6 where production is monitored via HMIs at the Master Control Station (MCS) and remote workstations that could be located further away in onshore offices.

Components of a Subsea Control System

This subsection highlights the core components of a subsea control system, their functions and general architecture. The functions of these components are described in Table 2.4. A subsea control system comprises of one or more of the following components [76]:

- a wellhead with connected casing strings;
- a subsea *christmas tree* comprising pressure and flow control valves;
- a production control and monitoring system for remote monitoring and control of various subsea equipment, possibly multi-phase flow meters;
- a chemical injection system (an equipment that allows injection of various chemicals into the reservoir fluid stream);
- an umbilical cable with electrical power and signal cables, as well as conduits for hydraulic control fluid and various chemicals to be injected into the produced fluid streams.

The components in the subsea architecture can be split into the following layers [6]:

1. **Hardware:** Sensors, actuators, RTUs, PLCs, server equipment (racks, CPUs), routers, access control hardware (smart cards, RFID, etc), and valves.
2. **Firmware:** Operating systems, data and instructions for controlling the hardware.
3. **Software:** HMIs, Application Programming Interface (APIs), proprietary software packages, and applications.

Component	Function
Wellhead Christmas Tree	Combines with wellhead to constitute the pressure barrier between reservoir and environment and allows for control of well through various valves and sensors
Subsea Electronic Module (SEM)	Collects sensor data from wellhead interfaces
Subsea Control Module (SCM)	Houses the SEM and control valve module
Umbilical Cable	Houses a collection of hydraulic, data (fibre optic), power cables
Master Control Station (MCS)	Main field control station where HMIs and servers are located for logging and processing real time system data
Topside Junction Box	Combines all electric and hydraulic power generated topside and transmits to subsea network (umbilical termination unit)
Hydraulic Power Unit (HPU)	Power source for hydraulics to move valve actuators
Electrical Power Unit (EPU)	Power source for electrical components

Table 2.4: Functions of Some Components of a Subsea Control System

4. **Network:** Communications protocols, modems/routers, firewalls

5. **Process:** Designed ICS business logic, control systems configuration

Attack Vectors of Subsea Control Systems

In this subsection, we introduce some attacks that a subsea control system could be vulnerable to, based on the integration and functions of its components as discussed earlier. The attacks proposed here and potential mitigation strategies highlight the dangers of the current system architecture used in controlling the steady production of volatile hydrocarbons from subsea to topside.

1. **Interception of Commands and Sensor Readings:** The initial stages of an attack requires gathering information on the system and operating parameters. This could be executed with a Man-in-The-Middle (MiTM) attack, where the connection between source and destination ports is intercepted, creating two new channels of communication: one connection between the source device and attacker, and another one between the attacker and the destination device [77]. This attack could target the software layer of the subsea architecture through the industrial network. Assuming an attacker managed to compromise a workstation within the MCS, they

would gain access to the HMI and sensitive information like pressure and temperature values, production flowrates, maximum allowable working pressure (MAWP), and valve fail-safe positions for example. Additionally, the attacker is able to act as a proxy and therefore read, insert, and modify data in the intercepted communication [77]. Earlier analysis has shown documented cyber-attacks involving theft of operational information as indicated in figure 2.5. Adding authentication and encryption of data helps to defend against this kind of threat. Park and Kang [78] proposed a solution to MiTM attacks by authenticating inter-device communication where each sensor is involved in the generation and distribution of session keys [74].

2. **Injecting Falsified Sensor Data:** The goal of this attack is to compromise the integrity of the sensor readings. This is a spoofing attack which is a variant of the MiTM attack where the attacker modifies data between two communicating devices. The firmware and hardware layers of the subsea architecture are typical targets of this kind of attack, which is executed through the industrial network. An attacker intercepting communication between the SCM and the MCS conveying sensor readings could modify these values even before the gateways (serial-to-ethernet converters) convert the data to ethernet packets [79]. Another example is where an attacker, using the compromised workstation, manages to modify control logic of the SCM altering upper or lower limits of set pressure points which could cause a well blowout. The oil spill in the Gulf of Mexico in 2010 has shown how devastating the impact of a subsea well blowout can be to the environment [80] and safety of personnel. Figure 2.5 shows six incidents of documented cyberattacks each involving modification of control logic and changing program state which indicates that threat actors have this capability. A number of studies have suggested mitigation against this type of attack by using physics-based methods [81] which consider the effects of the attack on the controlled physical process and look for deviations from expected physical sensor measurements [82]. Azzam et al. [82] proposed an Early Warning System (EWS) that, on its own, is not capable of detecting injection of false sensor readings, but can generate early warnings in ICPS

based on preliminary indicators. They applied their framework to Linear Time-Invariant (LTI) systems and adapted existing reachability analysis tools to compute a suspicion metric. This could prove useful if integrated with other intrusion detection capabilities to thwart stealthy malicious attempts.

3. **Denial of Service (DoS) Attack:** One of the most common attacks for cyber adversaries to conduct is the DoS attack [83]. System availability is of utmost importance in a subsea control system architecture and the attacker can flood the communicating device with requests to jam the communication channels and prevent legitimate requests [83]. DoS can compromise the network and hardware layers of the subsea control system and render engineers in the MCS incapable of sending emergency shutdown commands to shut-in wells discovered to be operating in unsafe conditions. Figure 2.5 shows reported cases of DoS attacks, in six instances, and loss of availability, in four instances, affecting upstream O&G facilities. DoS can have very serious impact by disabling critical equipment in a subsea control system architecture. Sicari et al. [84] proposed a defence mechanism against different types of DoS attacks named REATO. They examined a cross-domain and flexible middleware, named NetwOrked Smart object (NOS) and tailored REATO to it.

Overall, with many of these potential attacks on subsea systems, the pathway to initial compromise is the industrial network. As a result, it seems logical to conclude that alongside other mitigation strategies, a robust network monitoring and detection solution may significantly improve the security posture of these subsea systems. This is important because with the growth in offshore E&P activities due to rising number of mature (depleted) onshore oilfields in recent years [85] subsea production is set to dominate a significant market share in the industry. The major vendors in the subsea control equipment market are Subsea 7, Technip FMC, Akastor ASA, Baker Hughes, and National-Oilwell Vargo Inc [85] while for DCS we have ABB, Emerson, Honeywell, Rockwell Automation, Schneider Electric, and Siemens [86] dominating the market share. In isolation, these equipment are robust and are safe for operations. However, in a bid to increase their market share, these key vendors controlling the market share are designing products with more and more integration with corporate IT systems which introduces additional attack vectors with an increased risk of zero-day attacks.

2.4.3 Commonly Used Industrial Protocols in Oil and Gas

By far the most widely used communication protocol in the oil and gas industry to control and monitor operations is ModbusTCP/ModbusRTU [87, 88, 89], while others are OPC DA (Open Platform Communication Data Access) and EthernetIP. Industry data shows that these producing assets, due to their age and legacy systems, have continued to use these insecure protocols like ModbusTCP for critical communication. Replacing the entire communication protocol with a more secure one will be a hugely expensive endeavour. Moreover, the down time associated with such critical operations will not likely be acceptable to oil and gas management boards. Therefore, it is critical that researchers devise means to protect the current state of oil and gas assets. Availability of these systems is key and the communication of both control signals and sensor monitoring data are often not encrypted and not signed for data integrity [90]. This leads to the second research question:

RQ2: To what extent can vulnerabilities in prevalent communication protocols employed within the oil and gas industry be exploited further?

To answer this question, an oil and gas testbed utilising one of the common communication protocols (i.e. ModbusTCP due to cost and ease of implementation) would have to be designed and built to collect data. This research provides the basis for the design and installation of a gas wellhead monitoring testbed that emulates an oil and gas sub-system capable of providing the necessary data for further studies. To do this effectively, it is important to highlight historical cyber attacks on upstream oil and gas assets and look at what motivates the threat actors.

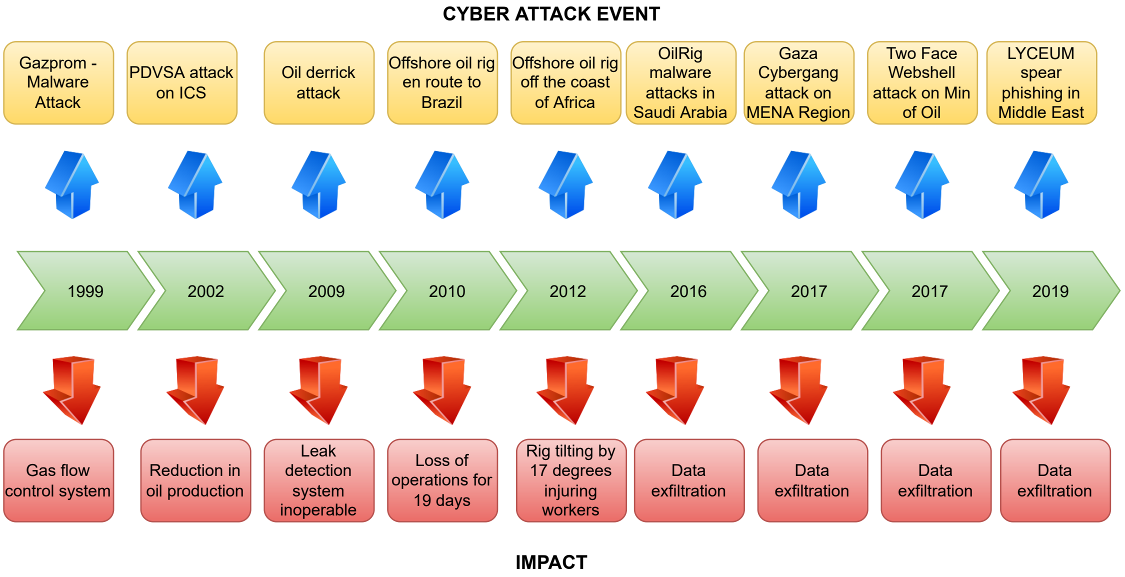

CYBER ATTACK EVENT
Gazprom - Malware Attack
PDVSA attack on ICS
Oil derrick attack
Offshore oil rig en route to Brazil
Offshore oil rig off the coast of Africa
OilRig malware attacks in Saudi Arabia
Gaza Cybergang attack on MENA Region
Two Face Webshell attack on Min of Oil
LYCEUM spear phishing in Middle East
1999
2002
2009
2010
2012
2016
2017
2017
2019
Gas flow control system
Reduction in oil production
Leak detection system inoperable
Loss of operations for 19 days
Rig tilting by 17 degrees injuring workers
Data exfiltration
Data exfiltration
Data exfiltration
Data exfiltration
IMPACT

Figure 2.7: Timeline of cyberattacks on Upstream Oil and Gas Facilities [6]

2.4.4 History of Cyber-Attacks on Upstream O&G Assets

A number of cases have been reported where upstream systems were directly or indirectly compromised by malicious insiders or malware, causing a number of adverse effects on operations and machinery [58], [6], [91]. Stergiopoulos et al. [6] catalogued 24 major cybersecurity attacks and events on upstream systems. We have used this information as a baseline to present a timeline of chronological security incidents that have affected the upstream O&G sector (see Figure 2.7). An interesting observation from the temporal characteristic shows a growing frequency of data exfiltration attacks against upstream O&G production companies in recent times which could be indicative of the consequences of increasing integration of real time OT monitoring parameters with corporate IT networks to improve decision making. This is a part of the digital oilfield trend being witnessed in the industry. The O&G industry, in particular, is very competitive and almost any kind of leaked information can be beneficial to a competitor [90]. Obtaining sensitive data like well drilling techniques, data on suspected oil and gas reserves, and special recipes for premium products [90] including chemical injection and corrosion inhibitors can prove to be very valuable and therefore attractive to attackers.

2.4.5 Threat Actors and Motivation

Threat actors operating in this sector typically range from those looking for ransom to those operating for rivals within the industry or outside and last (but not the least) state-sponsored agencies with specialised hackers at their disposal. The last category has immense resources and the potential to devastate critical infrastructure is massive [92] as seen in the case of Stuxnet - A virus that was reportedly designed by State intelligence to spy on and disrupt Iran's nuclear enrichment centrifuges, but also ended up spreading to infect Chevron facilities [93], a major O&G company.

They can generally be classified as [94] [95] [90]:

- Disgruntled Ex-Employee: Usually motivated by revenge on employer by triggering information disclosure to public to cause embarrassment, or to sell sensitive information. Person may still possess knowledge of sensitive information like passwords or system architecture.

- Insider Threat (Disgruntled Employee): Insider threat could also be motivated by revenge although there are several factors that could cause a person to turn against their employer. Defence against this kind of threat actor is very complex, as they have access to a lot of data.

- Hacktivists: This group is motivated by certain ideologies and will not hesitate to infiltrate a company they feel has gone against those principles. Their goals are usually to expose secrets and whistle-blowing.

- Nation State Hackers: These are hired by a Government to perform cyber operations against other nations. O&G producing nations usually rely on the revenue generated from oil production as a major source of economic power. This is what makes the impact of successful attacks to be significant to victim States. These groups are highly resourceful and aim to inflict maximum damage (loss of life and damage to environment).

- Cyber Terrorists/Organised Crime: Non-State hackers are groups or individuals with the main intention of obtaining money by stealing sensitive data or confidential information and either selling it or blackmailing the company into paying a ransom.

There are also some known adversaries that have been identified to be targeting the O&G sector. These include [29]:

- XENOTIME: This group has been known to target O&G companies in the United States and Europe since 2018 and have compromised several ICS vendors and manufacturers.
- MANELLIUM: Since 2013, this group has been targeting petrochemical companies.
- CHRYSENE: Involved in the 2012 Shamoon cyberattack at Saudi Aramco and remains active and evolving in more areas.
- HEXANE: Capabilities of this group is still being studied by Dragos but was first identified in 2019.
- DYMALLOY: A highly aggressive and capable activity group that has the ability to achieve long term and persistent access to IT and OT for intelligence collection and possible future disruption events.
- APT33: A group that has compromised oil companies in the United States, Europe, and Asia by obscuring a dozen live C&C (Command & Control) servers that have been used to do reconnaissance and botnet management since 2018 [90]. C&C connections to cloud services are difficult to detect since they use normal services that any employee could use for legitimate purposes [90].

In the next section, we will examine the unique challenges in the upstream O&G subsector that have made it attractive for these threat actors to actively carry out attacks on the targets discussed.

2.5 Challenges in securing upstream assets

There are some unique traits that make the upstream O&G sector more challenging to secure [29] [90] when compared to other critical infrastructure industries. These are highlighted as:

- Upstream assets are usually spread over a huge geographical landscape, including significant assets offshore.
- Offshore assets are usually in remote locations and in deep waters.
- A large percentage of production facilities have been designed decades ago and lack modern security features which make them vulnerable and obvious targets for cyberattacks.
- The frequent integration of vendor systems with operating company systems.
- Dependencies: Large distances and deep waters make it costly to establish a computer network for offshore platforms. Frequent damage to fibre-optic cables on the seabed makes it challenging to establish redundant and completely independent network solutions.

Accuracy is also a big challenge in oil and gas as the exact amount/volumes of what is produced is not easily measured [62]. Hydrocarbon volumes fluctuate depending on the environmental temperature and pressure conditions and require complex conversion calculations of the observed volumes at each custody transfer point [62]. It is possible to spoof this data in a way that will make it difficult to investigate [62]. Micro fractional changes to any one of the sensor parameters used in calculating hydrocarbon volumes over time could lead to significant losses to either operating companies or oil producing States. The latter could be better described as economic sabotage.

Process states and plant configurations are always changing, sometimes due to optimisations, but mostly as a result of degradation. An example is pressure vessels that have corroded beyond minimum thickness and can no longer withstand the Maximum Allowable Working Pressure (MAWP). Rather than outright replacement, vessels can be derated to a lower MAWP. This changes plant configuration and set point limits. Also, as discussed earlier, the life of a field in production may last up to 50 years, yet the assets have a design life considerably less than that - usually 25 years [96]. Life extension projects are carried out on facilities to extend, upgrade, and further optimise operations. This is why after a major maintenance phase, it is not unusual to have a system operating in a manner slightly different from prior to the maintenance activities. These configuration changes need to be taken into account when designing an efficient cybersecurity mitigation strategy.

The challenges in securing ICPS from cyberattacks in the offshore O&G industry can also be broadly grouped into operational, financial, and legislative.

Operational Challenges: Keeping OT running at all times is critical for any successful industrial plant: every second systems are offline can cost the operating company thousands of dollars and recovering from a single hour offline can take days [97]. To put this into an O&G context, assuming an average oil price of $70 per barrel, a facility producing 250,000 bopd (barrels oil per day) would be losing approximately $17,500,000 USD for each day of shutdown. This calculation does not yet take into account other operational costs or potential long term impacts which could increase the cost of the shutdown. An example of a field with such capacity is Nigeria's Agbami operated by Chevron. With such huge operational costs, it is even more critical for O&G companies to keep production going at all cost.

It is also not unusual to witness systems running without being patched for years because operations availability and system uptime have a higher priority than security within ICS environments [98]. In restarting production wells after a shut-in, producers must also weigh the cost and mechanical difficulty of restoring those wells back to pre-curtailed volumes [99] as the transfer of fluids back to the wellbore after production restoration is not usually very efficient or complete [100]. This creates a high-risk scenario where after a significant shutdown, depending on the age of the well, previous production levels may never be attained again. The industry is intolerant of frequent shutdowns and as a result, there aren't many opportunities for security updates and patches which are necessary as most of the offshore platforms are legacy systems. There is a high number of old offshore platforms still producing today. In fact, nine of the world's longest-standing fixed offshore platforms are located in the North Sea, while one is in the Gulf of Mexico, US [101]. One of the oldest of them is a platform called Ekofisk 2/4 B, operated by ConocoPhillips and located 2.3km north of the Ekofisk Complex in the North Sea and it has been operating since 1974! To keep these platforms running efficiently, the operators retrofit new technologies onto legacy systems. This is usually done without security considerations that would adequately protect these systems from cyberattacks.

As discussed earlier, drilling campaigns are undertaken throughout the life of a field. Whenever a service company is contracted to drill wells for an operating company, this requires the use of shared computer networks, resulting in production equipment being exposed to network-related vulnerabilities [29]. The frequent integration of vendor systems with operating company systems is another risk factor that increases the attack surface of O&G production platforms. There is a need to ensure all sub-contractors keep the same or a higher level of cyber-hygiene than the operating company.

Financial Challenges: The offshore O&G industry is a highly regulated and capital intensive industry [2]. For example, FPSOs (Floating, Production, Storage, and Offloading vessels), because they give operators the freedom and versatility to explore remote areas and extract at a significantly cheaper cost [102], have become very popular in exploring deep offshore. The cost of a typical FPSO could range from $800 million USD (Exxon Mobil's Kizomba A [103]) to $3 billion USD (Total Nigeria's Egina FPSO [104]). The upstream life cycle discussed earlier shows that the company bears these huge costs for a number of years (during the exploration, appraisal, and development phases) before production begins, and thus are trying to recoup huge investments made as quickly as possible during the production phase. However, despite the fact that the production phase is the most vulnerable to cyberattacks (because it is the longest and most active phase), the company's focus during this phase is to try to break even, turn in a healthy profit before the reservoir is depleted (limited period), and meet their obligations to the host Government through payment of taxes and royalties. To achieve this, continuous operations with minimal shutdowns are usually prioritised over security concerns.

Legislative Challenges: Several governments all over the world are starting to recognise the threat that cybersecurity poses to the critical infrastructure industry. Even though it is usually the norm that each individual company bears direct responsibility for securing its digital systems, severe cyberattacks will have national implications as well [105]. This means that governments and the relevant agencies have a role to play in detecting, preventing, and responding to such attacks [105]. A seamless transition between private sector companies to authorities will require a holistic threat picture, clear areas of responsibilities, and good procedures that are exercised regularly. This is hardly the case today [105]. In the United States, for example, there are stricter cybersecurity regulations that govern power, chemical, and nuclear facilities, but no federal laws impose such standards on the O&G industry [29]. O&G companies are not required to report cyber incidents, and as a result, the specifics are usually kept secret because companies tend to disclose information in exchange for anonymity [29]. This ensures that lessons learned from cyberattacks in one company and security measures implemented in response to such attacks are not always passed on to other companies in the sector, creating a serious knowledge gap [29]. Attacks are getting more sophisticated and government legislation is playing catch up. There needs to be a concerted effort to create a legislative framework that ensures a minimum requirement for companies to secure their critical infrastructure assets from cyberattacks.

The financial and legislative challenges have only been highlighted to give context to the bigger industry problem, however neither are within the scope of this thesis – which will be focused solely on the operational challenges. The next section will identify some general mitigation strategies, and how current datasets available to OT security researchers are inadequate for the O&G industry.

2.6 Securing Upstream O&G Assets - Current State

As elaborated earlier, the O&G industry is increasingly transitioning towards Internet of Things (IoT) technologies and digitalisation, aiming to enhance efficiency, safety, and operational insight. However, with further integration of advanced sensors, data analytics, faster communications utilising ethernet-based protocols, these advancements also expose the industry to new vulnerabilities. It is therefore useful to highlight some of the general mitigation strategies available to reduce this exposure.

2.6.1 General Mitigation Strategies

General cyber security safeguards such as restricted physical access, cryptography, patch management, separation of corporate and production systems (through Demilitarized Zones (DMZ), Firewalls and Access Control Lists (ACLs)), and activity logging are all applicable mitigation strategies, but need to be viewed in conjunction with typical SCADA systems characteristics [52]. Although very little has focused on O&G assets, in the broader context there are some practical applications that can improve the cyber hygiene of upstream assets. Esfahani et al. [106] and Srinivas et al. [107] are both studies that proposed the use of lightweight authentication to ensure only authorised users gain access. In [106], a Machine-to-Machine (M2M) protocol based on hash and XOR operations was applied in two phases - (a) the registration phase, where each smart sensor registers itself to an authentication server with replication of pre-shared keys with the router, and (b) the authentication phase where mutual authentication is achieved between the sensor and the router [74]. [107] was based on chaotic map for IIoT environments which allows access to designated IoT devices only to authorised users with the use of personal biometrics, smart cards, and passwords.

Research on ensuring basic security or defending against dreadful attacks in IIoT is still in its infancy [74] especially for the O&G sector, however, in the next sub-section, we shall examine intrusion detection systems and the limitation of datasets available to expand security research in this area that is applicable to the sector.

2.6.2 Intrusion Detection Systems (IDS)

Intrusion detection Systems (IDS) can be classified broadly into misuse detection (signature/rule-based Intrusion Detection Systems) and anomaly detection [108]. Historically, misuse IDS have proven to be effective in identifying traditional (known) cyber attacks that indicate discriminate patterns [109, 110, 111, 112], but in spite of this, these IDS are less effective at detecting zero-day attacks that utilise novel methods of exploiting vulnerabilities with persistence. This is an advantage that anomaly detection methods have because they are more capable of detecting novel anomalous scenarios. However, despite this advantage, anomaly detection solutions are not commonly applied in practice because of high computational overheads and high false-positive rates [113] - often leading to a high number of false alarms overwhelming security experts with alerts [114]. This is critical, since the usefulness of intrusion detection systems is greatly influenced by the false-positive rate [115].

With the increasing frequency of zero-day attacks being carried out on critical infrastructure [116], it has become evident that to improve widespread adoption, anomaly detection methods need to be improved upon to reduce the high false-positive rates and computational complexities. This would potentially increase the protection levels of critical infrastructure against cyber threats.

The lack of adequate datasets remains a huge challenge to security research in this area. Machine learning (ML) has been used for the identification of anomalous behaviours in industrial and manufacturing systems [117]. An ML-based firewall suggested by Haghighi et al. [118] towards securing ICS was focused on accuracy and achieving zero false positives in developed classifiers. In another example, Anthi et al. [119] explored how adversarial attacks can be used to target supervised classifiers by presenting generated adversarial DoS samples to a trained model and understanding their classification behaviours on IoT devices.

Bhamare et al. reviewed related works in the field of securing ICS/SCADA from cyber threats using machine learning, summarised in Table 2.5 [53]. The studies were however limited in the scope of application as most were from specific industry data sets or limited simulated models of ICS that are not applicable to the O&G industry (shown in Table 2.6). Further studies carried out by Zeng et al. [120] introduced a taxonomy of detection approach and also discussed machine learning-based solutions along with other types of available approaches for IDSs deployed in ICS [120] [53]. By their own admission, the authors confirmed that from the papers they surveyed, power systems are the main field that investigators study in [120]. From their comparison, most of the datasets or testbeds utilised were from power grids and a small percentage from water distribution systems.

Ref.	ML Model and Implementation
Wehenkel [121]	Decision tree induction, multilayer perceptron and nearest neighbour classifiers
Dua and Du [122]	Cybersecurity using ML and data mining in general
Cardenas et al. [123]	Attack categorisation, IDS
Zhang et al. [124]	Support Vector Machine (S2 OCSVM), IDS
Yasakethu & Jiang [125]	Artificial Neural Network, Support Vector Machine, Hidden Markov Model
Beaver et al. [126]	Anomaly detection in SCADA via comparison of various ML algorithms
Maglaras & Jiang [127]	One class Support Vector Machine, IDS
Hink et al. [128]	OneR, NNge (Nearest Neighbour-like algorithm), Random Forests, Naive Bayes, SVM, JRipper, Adaboost
Erez & Wool [129]	Single window classification algorithm deployed on IDS to detect irregular changes in SCADA control register values
Franc et al. [130]	A Multiple Instance Learning algorithm used on network logs for security
Nader et al. [131]	ML techniques with kernel methods to detect cyberattacks in water distribution systems
leahy et al. [132]	Classification ML techniques
Valdes et al. [133]	Unsupervised ML methods for anomaly detection in electrical substation circuits
Stefanidis & Voyiatzis [134]	Hidden Markov Model, IDS
Bartos et al. [135]	Support Vector Machine-based classification system

Table 2.5: ICS/SCADA Cybersecurity: Summary of Machine Learning Approaches [53]

ML Technique	Authors	Domain Secured
SVM/OCSVM	[126] [136] [137] [138] [125] [139] [127]	integrity, availability, confidentiality
Naïve Bayes	[140] [126]	integrity, confidentiality
Decision Trees/Random Forests	[126] [140] [141]	integrity, confidentiality
Deep Belief Network	[136] [137]	availability, integrity
Artificial Neural Network	[136] [125]	integrity
KNN/K-means	[142] [143]	authentication, confidentiality, availability, integrity

Table 2.6: Popular ML techniques used in ICS Security

Challenges in the way of utilising machine learning and how it can help in defence mechanisms with respect to the relevant threats in ICS have been reviewed comprehensively by Zolanvari et al. [144], [53]. A case study was also presented where an ML-based IDS was developed using a SCADA testbed. The dataset from the testbed was deliberately built to be imbalanced by making the percentage of attack traffic in the dataset less than 0.2%.

A comparative analysis of various ICS datasets, summarised in Table 2.7, was carried out by Choi et al. [145]. The analysis seems to agree with our observed limitations of the current datasets used to conduct ICS security research and highlights why most are not applicable to a broad set of scenarios. For our case specifically (O&G offshore industry), most of the datasets do not account for the dynamic behaviour of monitored variables identified earlier. Pressure and temperature values change throughout the life of a producing field as the reservoir is depleted which results in different hydrocarbon volumes calculated at any point in time. The monitored variables in current datasets only fluctuate within a given range. This is summarised in Table 2.8.

The review of existing literature shows that although a lot of research has been conducted on IDS security, the common limitation has been the availability of a wide-scope dataset that applies to several critical infrastructure industries. The power industry is the most represented sector – which presents an opportunity to create new datasets that represent commonly deployed ICS setups in the O&G industry.

In addition, despite extensive research in attack detection, the dynamic nature of attackers constantly evolving their tactics necessitates ongoing studies to stay ahead of new threats

ICS Dataset	Protocols	System	Year of release	Data-type	Reference
Morris et al.	Modbus	Power, water, gas	2013, 2014, 2015, 2017	csv, arff	[146]
Lemay	Modbus	SCADA sandbox	2016	csv, pcap	[147]
SWaT	Modbus, Ethernet/IP	Water treatment	2016	csv	[148]
Rodofile et al.	S7Comm	Mining refinery	2017	csv, pcap	[149]
4SICS	Modbus, S7Comms, DNP3, Ethernet/IP	Complex	2015	pcap	[150]
S4x15 ICS Village CTF	Modbus	Complex	2015	pcap	[151]
DEFCON 23 ICS Village	Modbus	Complex	2015	pcap	[152]

Table 2.7: Summary of ICS datasets publicly available [145]

and vulnerabilities. The increasing popularity of ML-based intrusion detection systems applied to some select sectors of critical infrastructure security motivates the third and fourth research questions:

RQ3 – How well do supervised machine learning methods fare in identifying cyber-attacks against a typical oil and gas OT setup?

				Data Capture		
ICS Dataset	**Num. of Pkts**	**Byte of Pkts**	**Duration**	**Continuous**	**Interruptions**	**Dynamic Variables**
Lemay	2,588,491	169,690,458	15 hours	No	Yes	No
SWaT	19,761,714	5,498,545,489	11 days	Yes	No	No
Rodofile	23,387,064	5,848,801,728	27 hours	Yes	No	No
4SICS	3,773,984	314,562,089	1d 22 h 7m	Yes	No	No
S4x15CTF DEFCON23	1,678,668	124,271,095	N/A	Yes	No	No

Table 2.8: ICS Datasets: Data capture summary

RQ4 – How efficiently can unsupervised machine learning methods be utilized to detect anomalies in industrial cyber-physical systems?

To answer these questions, this research will investigate anomaly detection IDSs that use machine learning algorithms for pattern recognition to detect threat activities that are anomalous for a particular system, and other IDSs which use signature/misuse-based systems to compare the activities to a database of known threats [125], [127], [122], [53].

2.6.3 High Periodicity of Industrial Communication Networks

Network traffic from industrial networks exhibits strong periodic patterns [153]. This is because, rather than traffic being generated mainly from random user-generated workflows - as in the case of enterprise/IT networks - industrial network traffic is primarily generated from the consistent polling of data between systems with the aim of monitoring and controlling the process. This gives it a high repeatability resulting in a consistent pattern. This could be likened to a heartbeat, where each industrial network has its own rhythm represented as a pattern. This pattern is a basic representation of the behaviour of the industrial network under normal operations.

Having such high periodicity has its advantages. One such advantage is that, if properly represented and modeled, any slight deviation from the established basic network pattern could be easily identified and flagged as anomalous behaviour – similar to diagnosing an irregular heartbeat. This regularity of patterns present in industrial control network traffic makes anomaly detection very promising [154, 17].

Moreover, it is noteworthy that despite their proficiency in identifying zero-day attacks, anomaly detection methods are not widely adopted within conventional IT networks, primarily due to the highly dynamic nature of enterprise network environments. However, the predictable traffic patterns of an industrial network makes the prospect of anomaly detection highly promising [17]. This is investigated in the fifth and sixth research questions:

RQ5 – How can the inherent periodicity within industrial networks be effectively leveraged to enhance anomaly detection?

RQ6 – Is the investigated unsupervised anomaly detection method adaptable across different industrial networks?

2.7 Conclusion

This chapter examined the growing threat of cyberattacks to ICPS in the offshore O&G industry as a result of the advancements in technology, digitalisation, and integration of oil field equipment with corporate networks, and the need for remote monitoring and control. This has increased the attack surface available for attackers to exploit. A timeline of documented cyberattacks on upstream O&G assets was presented which showed that data exfiltration has become more common in recent times, coinciding with an increase in the integration of OT equipment with IT networks that is now prevalent in the O&G industry. An overview of offshore O&G operations and the associated production process is also described, highlighting potential areas where cyberattacks may originate.

A typical subsea control system architecture was also analysed and its vulnerabilities to MiTM, DoS, and spoofing attacks by mapping the attacks to one or more layers of the architecture were highlighted. Correlating these to reported cyber security incidents that affected the upstream O&G industry in recent times showed that threat actors have the capability to breach subsea control systems in its current state. We also discussed challenges in securing upstream assets, highlighting dynamic process state changes due to operations like de-rating of pressure vessels and asset life extension projects which add to the complexity of identifying whether a changed plant configuration is legitimate or due to malicious actors. Mitigating strategies were also highlighted involving the use of IDS. There remains a lack of adequate datasets representative of processes in upstream oil and gas production.

The remainder of this thesis focuses on the improvement of detection capabilities of IDSs by investigating ML (supervised and unsupervised) methods, analysing the detection models, and establishing a baseline for the development of a novel anomaly detection model. To achieve this, new datasets that apply to the OG industry will be required. This data would be collected from a scaled-down version of an OG sub-system testbed purposefully designed and built to aid this research.

Chapter 3

Research Methodology and Approach

In Section 2.6.2, we have explained that contemporary advancements in the area of cyber attack detection in industrial environments predominantly revolve around the utilisation of ML-based intrusion detection systems. Nevertheless, it is noteworthy to highlight that these models have primarily found their application and development in other critical infrastructure domains, such as power grid systems, with comparatively limited attention devoted to the Oil and Gas industry.

In this chapter, we present a research methodology for exploring the application of ML techniques (supervised and unsupervised), and a further method focused on measurement around time, to harness the intrinsic high periodicity exhibited within industrial networks. Prior research has thus far under-explored this critical aspect, presenting an opportunity to address this research gap effectively.

The primary focus is on developing and evaluating anomaly detection methods that are tailored for industrial network environments, particularly within the context of networks exhibiting high periodic behaviour. As a result, deep learning algorithms, while highly effective and versatile, shall not be included in this research. This is because deep learning methods often demand substantial computational resources and data, which may not be readily available in such settings. To ensure the practical applicability and relevance of the findings, we have opted to concentrate efforts on traditional supervised and unsupervised ML methods, while exploring a further option which focuses on the high periodicity of industrial network communication. The spread of these methods provides a balance between efficiency, accuracy, and interpretability, making them more suitable for the industrial network cybersecurity challenges we aim to address.

To aid these experiments, a testbed will be designed and built in line with industry standard principles and best practices which shall be explained in the following subsection.

3.1 Testbed Design: Gas Wellhead Monitoring Station

An industrial SCADA (Supervisory Control and Data Acquisition) testbed will be designed to emulate a gas wellhead monitoring station, serving as a representative subcomponent node within a subsea control system. The key components of the testbed included:

1. Remote Terminal Unit (RTU): Acting as the central controller, the RTU will facilitate the communication and control of the various sensors and actuators in the system.

2. Flow Meter (Air): This device will measure the air flow rate through the system, simulating the monitoring of gas flow in a real wellhead platform.

3. Pressure Sensor: Installed to measure the pressure within the system, providing critical data necessary for safe and efficient operation.

4. Temperature Sensor: A Resistance Temperature Detector (RTD), typically used in oil and gas systems, will monitor the temperature within the system, ensuring that operational parameters are maintained.

5. Air Compressor: This component will be used to introduce air into the system, creating the flow necessary for testing and emulating real-world gas flow conditions.

6. Shutdown Valves (Solenoid Valves): These valves are essential for emulating safety shutdown procedures, automatically stopping the flow in response to specific conditions to prevent accidents.

3.1.1 Configuration and Emulation

The configuration of the testbed will be guided by the architecture of a typical subsea control system as explained in Chapter 2. Each component will be carefully selected and positioned to reflect its real-world counterpart, ensuring that the testbed provides a realistic and practical environment for research and testing. The polling interval configured on the system conformed with industry best practices to ensure real-time data acquisition and control to maintain efficiency and safety in remote operations. The RTU coordinates the data from the flow meter, pressure sensor, and temperature sensor, while the air compressor maintains a controlled flow of air through the system. The solenoid shutdown valves will be configured to respond to the RTU's signals, demonstrating their role in safety protocols.

3.1.2 Ethical Considerations

The design and operation of the SCADA testbed shall adhere to the Menlo Report principles [155], ensuring a responsible and ethical approach to research involving information and communications technology (ICT).

- Privacy and Confidentiality: Measures have been implemented to protect the identity of Original Equipment Manufacturer (OEM) of the RTU/PLCs involved in experiments and the data generated by the testbed. These include obfuscating the PLC images and incorporating a generic naming convention (i.e. PLC 1, PLC 2, PLC 3). Data handling protocols will also be established to prevent unauthorised access to the testbed.

- Minimising Harm: The testbed shall be designed with safety features, including shutdown valves and emergency procedures, to minimise any risk to researchers and equipment. Additionally, the use of air instead of hazardous gases will further reduce potential risks.

- Compliance: The testbed design and operations shall comply with relevant regulatory standards. Components have been certified by OEMs according to industrial standards.

3.1.3 Safety Standards

The testbed shall adhere to the following safety standards:

- Component Certification: All components used in the testbed, such as sensors and valves, have been certified according to relevant industrial standards to ensure reliability and safety.
- Redundancy and Reliability: Critical systems, particularly those involved in safety shutdowns, will be designed with redundancy to prevent single points of failure.
- Emergency Procedures: Emergency procedures have been established, including manual override capabilities, to handle any unexpected situations swiftly and effectively.

3.2 Research Design - Experiments

The experiments to aid this investigation will employ a multifaceted research design that encompasses the following key elements:

3.2.1 Data Collection:

To conduct our analysis, we will collect data from the testbed designed as proposed in Section 3.1. This testbed shall be purposefully built for this research and will replicate an industrial environment, specifically, a gas wellhead monitoring station within a subsea control system operating in the oil and gas industry.

Data collection within the testbed will involve the utilisation of sensors and network traffic data, closely mirroring the conditions found in operational wellhead control systems. This approach ensures that the dataset we analyse is not only representative of industrial network behavior but also tailored to the behaviour of some aspects of oil and gas industry operations.

3.2.2 Exploiting Communication Vulnerabilities:

In addition to data collection, the research design will involve an examination of vulnerabilities in popular industrial communication protocols used in the offshore oil and gas industry. Based on the survey conducted in Chapter 2, it was identified that ModbusTCP is the most widely used communication protocol in this context. Therefore, we will focus on exploiting vulnerabilities specific to ModbusTCP to gain insights into potential security risks in real-world industrial applications. This effort aims to shed light on the security implications of this widely adopted communication protocol, further enhancing the knowledge base of vulnerabilities against it.

3.2.3 Preprocessing and Feature Engineering:

Prior to applying ML techniques, we will preprocess and engineer features from the collected data. This phase will include data cleaning, feature extraction, and the integration of insights gained from the exploitation of communication vulnerabilities into the feature engineering process.

3.2.4 Supervised ML Techniques:

In the analysis of supervised ML techniques for detecting cyber threats in industrial networks, we will apply a model selection process that considers various algorithmic characteristics. This approach allows us to tailor the selection to the specific needs of the industrial context. The model selection criteria will encompass the following elements, consistent with the relevant literature:

- Conditional Dependencies vs. Conditional Independence Models: This research will encompass supervised ML algorithms that function based on capturing conditional dependencies within the dataset, as well as those that assume conditional independence among variables.

- Discriminative Models: Within the realm of supervised learning, we will evaluate models that aim to maximize information gain without necessarily modeling the underlying probability or structure of the data. These discriminative models are invaluable for detecting nuanced and context-specific cyber threats in high periodicity industrial networks, where the data may not adhere to conventional statistical distributions.

- Ensemble Models: To maximise the predictive performance of the supervised ML models, we will incorporate ensemble techniques. Ensemble models amalgamate the outputs of multiple ML algorithms, each with its unique strengths and weaknesses, to achieve superior predictive results.

Combined, these model selection criteria ensure that we employ a diverse array of supervised ML techniques that cater to the requirements of industrial networks. This approach aligns with the evolving landscape of industrial cybersecurity, where a multifaceted response is essential to effectively combat a wide range of threats.

3.2.5 Unsupervised ML Techniques:

In parallel, we will explore a range of unsupervised ML methods to effectively uncover hidden patterns and anomalies within the industrial network data. Model selection criteria will be applied, considering the most frequently used and well-established approaches as follows, in accordance with the existing literature:

- Distance-Based Methods: This analysis will encompass unsupervised algorithms that primarily rely on distance-based metrics to measure similarities or dissimilarities between data points. These methods are invaluable for identifying anomalies and clusters within the dataset, particularly in scenarios where the distances between data points play a critical role in defining normal behavior and deviations.

- Model-Based Methods: We will investigate model-based unsupervised ML techniques that detect anomalies by constructing models to identify data points that deviate significantly from the majority. This typically involves learning complex models that capture the global distribution of the data and, as result, are designed to capture the underlying structure of the data by isolating outliers. This approach is particularly suited for identifying unusual patterns and deviations from expected behavior in industrial network contexts.

- Density-Based Methods: Another focus of this research will be on density-based methods, which aim to identify clusters or anomalies based on the density of data points in feature space. These methods are well-suited for scenarios where the data distribution may not be uniform, and anomalies are often defined by low-density regions.

An additional objective in these experiments is to assess the false positive rates of each of the investigated unsupervised ML algorithms. High false positive rates are a well-established limitation of unsupervised techniques, and we aim to identify which of the algorithms exhibit the lowest false positive alerts. This identification will also serve as a basis for the development of a novel anomaly detection model that can mitigate the issue of false positives more effectively.

3.2.6 Timing-Based Detection Techniques:

In this research, we will delve into timing-based detection techniques to capitalise on the high periodicity nature of industrial network communications. To achieve this, we will employ methods for measuring and defining the periodicity inherent in the communication patterns of these networks. These techniques will enable us to identify and characterize the temporal regularities and rhythms within the industrial network data, providing a foundation for the proposed anomaly detection methodology.

Furthermore, the components and insights obtained from the study of unsupervised ML detection will be leveraged in this phase. The results and patterns uncovered through unsupervised ML analyses can inform the development of novel anomaly detection models, which will specifically target the identification of irregularities in the temporal behavior of industrial network communications. The primary aim of this approach is to enhance the detection of threats while minimizing false positives.

It is essential to note that this research places a specific emphasis on low-complexity solutions that can be effectively deployed in industrial environments. As a result, deep learning algorithms, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, which are renowned for their exceptional performance in time series applications of anomaly detection, will not be included in the investigation.

3.2.7 Model Validation and Adaptability:

It is essential to ensure the reliability and adaptability of any model developed in the course of this research. Further testing shall be conducted on publicly available industrial datasets that encompass diverse industrial network verticals. The objective is to evaluate the model's performance and effectiveness in identifying anomalies in different critical infrastructure communication systems, leveraging the high periodicity nature common to such environments.

3.3 Conclusion

In this chapter, we detailed the design and development of an industrial SCADA testbed, which emulates a gas wellhead monitoring station. The testbed, composed of essential components such as the RTU, flow meter, pressure sensor, temperature sensor, air compressor, and shutdown valves, serves as a miniature representation of a node within a subsea control system. The configuration and operational protocols of the testbed were designed to reflect real-world conditions and functionalities, providing a robust platform for research and experimentation. We also underscored the ethical considerations and safety standards adhered to during the design and operation of the testbed, guided by the Menlo Report principles, contributing to the integrity of the research.

Furthermore, the chapter laid the groundwork for subsequent chapters that delve into the research design, specifically focusing on experiments involving data collection, machine learning techniques, timing-based detection techniques, and model validation and generalisation. These methodologies will leverage the realistic environment provided by the SCADA testbed to derive meaningful insights and advancements in the field of industrial control systems.

This research strives to contribute to the enhancement of cybersecurity within high periodicity industrial networks and, in addition, offer a versatile solution that can be leveraged across various industrial verticals, ensuring the utmost security and resilience in the realm of critical infrastructure communication.

Chapter 4

Field Flooding Attacks – Detection using supervised machine learning

Parts of this chapter have been published in the paper ”Detection and Mitigation of Field Flooding Attacks on Oil and Gas Critical Infrastructure Communication.” Computers & Security, 103007. https://doi.org/10.1016/j.cose.2022.103007

4.1 Introduction

In this chapter, the experimental setup of a gas wellhead monitoring testbed is described and its normal operational mode is compromised by exploiting the vulnerabilities in its communications protocol (ModbusTCP) using a novel field flooding attack. This novel attack is further evaluated using two additional testbeds to examine how different industry verticals behave when under this attack. Finally, eight supervised machine learning classifiers are evaluated to effectively detect this attack. Through doing this, the following research questions will be answered in this chapter:

- ***RQ2*** *To what extent can vulnerabilities in prevalent communication protocols employed within the oil and gas industry be exploited further?*
- ***RQ3*** *How well do supervised machine learning methods fare in identifying cyber-attacks against a typical oil and gas OT setup?*

In answering these questions, the following contributions are made:

- ***C2*** *Design and installation of a wellhead monitoring testbed to emulate the oil and* *gas production process and aid cybersecurity research in the OG industry.*

- ***C3*** *This research identifies a novel "Field Flooding" attack on the ModbusTCP protocol which can lead to a severe Denial of Service (DoS) attack.*

- ***C4*** *This research evaluates an Intrusion Detection System to effectively detect the Field Flooding attack on industrial control networks using a supervised machine learning approach.*

The remainder of this chapter is structured as follows; Section 4.2 discusses the modbus protocol structure and current state of detecting attacks in this research area. Section 4.3 describes the attack methodology, attacker model, and tools used including the testbeds utilised in the study. In Section 4.4 the results of the experiments are provided, while Section 4.5 analyses these results in more detail. In Section 4.6, supervised machine learning techniques are applied to detect the Field Flooding attack, and the performance of these techniques is evaluated. Key lessons learnt and a summary is included in Section 4.7.

4.2 ModbusTCP Protocol

The literature highlighted in Chapter 2, identifies that the most widely used protocol in the oil and gas industry is the ModbusTCP protocol. It was also shown that theft of operational information and Denial of Service (DoS) attacks are the most frequent impacts of documented cybersecurity incidents in the OG industry [156].

These incidents have led to a corresponding increase in security research focused on OT and critical infrastructure communications. However, due to the high cost of OT equipment, most research is carried out in simulated environments which may not represent exact OT system behaviour during cyber attacks. Consequently, not much is known about attack impact across different industrial environments. Would the same attack behave differently in a different industrial environment? These factors have motivated the study presented in this chapter with a focus on 1) the ModbusTCP protocol, 2) Denial of Service attacks, and 3) implementation of attacks on different real industrial systems to analyse behaviour/response to attacks.

More specifically, a novel Field Flooding attack is presented, which alters the structure of the ModbusTCP packet with additional malicious fields to target the PLC controlling critical processes. The attack involves sniffing network packets (Man-in-the-Middle) for ModbusTCP communications and injecting the malicious packets to the PLC to cause a denial of service. The Field Flooding attack is unique from most Man-in-the-Middle (MitM) and DoS attacks studied in the literature in the following ways:

- Does not require ARP poisoning as an initial step so would not be mitigated with standard measures capable of detecting ARP poisoning - a typical defence against MitM attacks.

- Does not increase the rate of packet transmission to the PLC (e.g. SYN Flood - a popular type of DoS attack widely studied). Rather, with much fewer, carefully crafted packets, can overwhelm the PLC which could prevent response to requests. This results in a behaviour that requires a different approach for detection/mitigation besides known measures (e.g. packet rate limiting).

4.2.1 Structure of the ModbusTCP Packet

The ModbusTCP protocol communicates using a simple request/ reply mechanism between a control centre and field devices [15]. The control centre(s) are the clients (formerly called ’Master’), while the field devices are the servers (formerly called ’Slaves’). This variant of the Modbus protocol uses TCP/IP as a transport mechanism for Modbus messages. There are four data storage modes in Modbus servers to store analog and digital input/output (I/O) which are highlighted in Table 4.1.

I/O Range	Description
00001 - 10000	Read/Write discrete output or coils
10001 - 20000	Read discrete inputs
30001 - 40000	Read input registers (16-bit registers for analog inputs)
40001 - 50000	Read/Write holding registers (16-bit storage)

Table 4.1: Modbus addressing format for data storage

A function code (FC) included in a Modbus message describes the purpose of the message [157]. Table 4.2 describes the most used public FCs by vendors while Figure 4.1 shows

Table 4.2: Most used public Modbus function codes

Function	Code	Hex	Type	Size (Bits)
Read Discrete Inputs	2	0x02	Read Only	1
Read Coils	1	0x01	Read/Write	1
Write Single Coil	5	0x05	Read/Write	1
Write Multiple Coils	15	0x0F	Read/Write	1
Read Input Registers	4	0x04	Read Only	16
Write Single Register	6	0x06	Read/Write	16
Read Holding Registers	3	0x03	Read/Write	16
Write Multiple Registers	16	0x10	Read/Write	16

the basic structure and size allocated to each header. The Modbus Application Data Unit (ADU) has a total size of 260bytes. This is shared by the Modbus Application (MBAP) header and the Protocol Data Unit (PDU) in the order of 7bytes and 253bytes respectively. The fields in the MBAP header are explained as follows:

- **Transaction ID:** This is a number that matches the Modbus server [Programmable Logic Controller (PLC)] response to its corresponding query from the Modbus client [Human Machine Interface (HMI)] and is incremented by one for consecutive queries.

- **Protocol ID:** This is usually set to “0” to indicate ModbusTCP protocol.

- **Length:** The length field indicates the size of the data (in bytes) in the rest of the packet (i.e. size of Unit ID, Function Code, and Data fields) so the receiving party knows what to expect from the packet.

- **Unit ID:** This is set to the Unit ID of the Modbus server the client wishes to communicate with. For the ModbusTCP protocol, the Unit ID is not relevant as the IP address of the server dictates the destination of the packet.

- **Function Code:** The function code identifies the action the Modbus server should take.

- **Data:** The Data field contains the data to write/ read and the address of the data stored on the Modbus server.

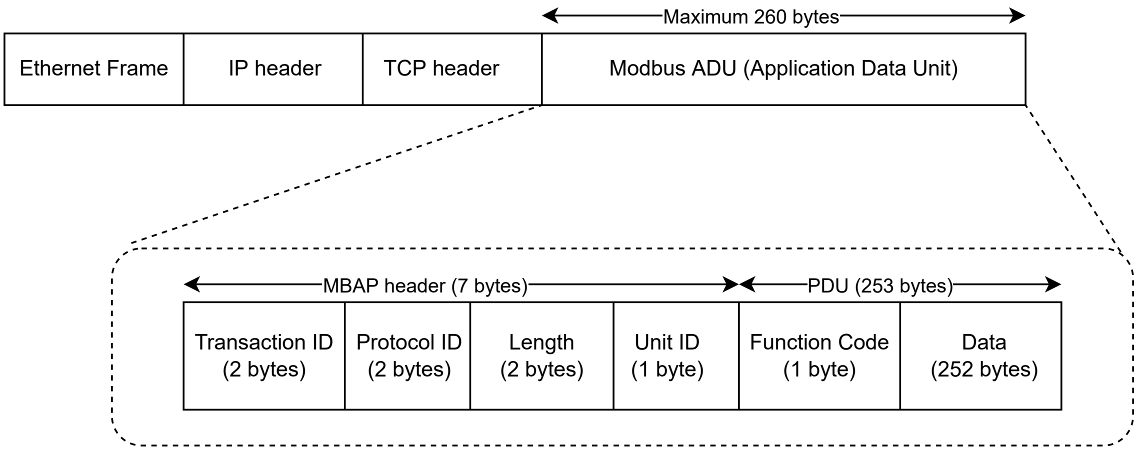


Figure 4.1: ModbusTCP packet structure

The Client-Server Query-Response Cycle: Queries from Modbus clients (e.g. HMI) and the corresponding response from Modbus servers (e.g PLCs) are sent in loops that are milliseconds apart. The query from the client contains the FC that tells the server what action to perform [158]. The "Data" field contains the address information that should be read or written to and specifies how many addresses to consider.

The corresponding response from the Modbus server (e.g. PLC) is usually an echo of the FC in the query [158], unless an error occurs. The data returned by the server indicates process status (in the case of a read request) or confirmation of data written (in the case of a write request). The packet structure of read and write queries/responses is shown in Figure 4.2.

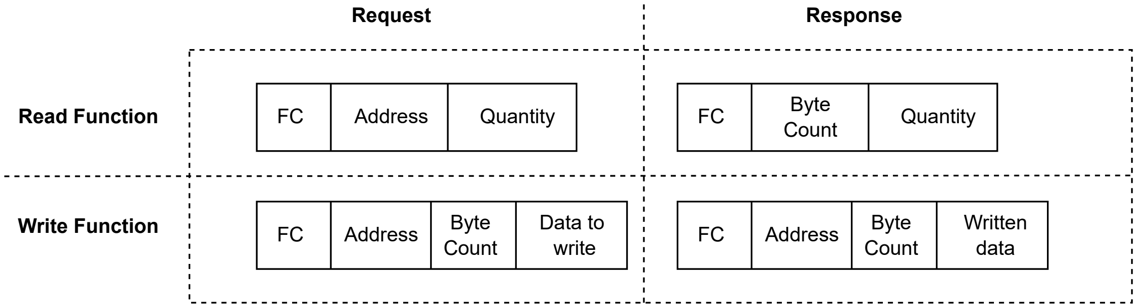


Figure 4.2: ModbusTCP message structure for memory access operations

4.2.2 Detecting attacks on the Modbus protocol - Current State

Vulnerabilities in the Modbus protocol have been widely considered, primarily due to lack of authentication and ease of deployment of this protocol. This section focuses on presenting relevant work that focuses on: (a) vulnerabilities reported in the ModbusTCP protocol – these studies have been carried out mostly on simulated testbeds, and (b) studies focusing on Intrusion Detection Systems (IDS) for the ModbusTCP protocol.

ModbusTCP Vulnerabilities: Chattha et al. [159] presented an implementation of cyber-physical systems with ModbusTCP communication for real-time security testing. Their study used two simulated case studies (i.e. Automatic Voltage Regulation and DC motor position control) using MATLAB Simulink, OpenPLC, and ScadaBR to understand the effects of attacks launched on the system. The authors of [160] used a penetration testing approach to identify attacks on SCADA systems specifically focusing on the ModbusTCP protocol. Their study combined three simulation tools (i.e. Qmod master, Modbuspal and, Conpot server) that were utilised for attacking the ModbusTCP protocol and developing countermeasures. Similarly, Parian et al. [161] carried out two attacks on the ModbusTCP protocol comprising of a MitM and malware attacks where the latter involved modifying requests made by the Modbus client, ensuring that the response from the server is reversed. They utilised `Scapy` (tool discussed further in Section 4.3.1) to manipulate the Modbus server response by changing the value of the requested coil. Our attack approach in this study however utilises `Scapy` differently to alter the ModbusTCP packet structure rather than change the value of the Modbus command/response. Their experimental setup was based on virtualisation technology, with the client, server, and attacker machines all hosted within Virtual Machines. However, a key limitation of the aforementioned studies is that they are all based on simulated environments which do not fully reflect real system usage [162]. These studies, therefore, did not consider attacks that alter the ModbusTCP packet structure and did not evaluate the impact of the attacks on a physical industrial testbed.

Furthermore, Bashendy et al. [163] presented a formal attack tree for representative explored attacks against the ModbusTCP protocol that models the attack steps in detail with different attributes. They categorised the attacks using the CIA triad (Confidentiality, Integrity, and Availability) where various modifications of the packets are made. Modifications included changing the FC to an unsupported one, injecting a replayed payload, or changing a specific value in the payload [163]. Similarly, in [164], the authors also highlight the vulnerability of the ModbusTCP protocol to malicious attacks using standard attack tools utilised in penetration testing. The attacks carried out in their study which impacted the system were limited to data manipulation (writing coils), MitM, and DoS. These studies, however, did not consider attack vectors dealing with protocol mutation by altering the ModbusTCP packet structure. Finally, Alcaraz et al. [165] explored security issues related to covert channels applied to ModbusTCP in industrial networks using a testbed comprising of various equipment including a Raspberry Pi 3 board simulating the logic of a PLC. They presented two approaches based on 1) timing - where insignificant delays are injected in the TCP/IP channels, and 2) storage - by the inclusion of hidden data in specific fields of the ModbusTCP packets. While the attacks presented in these studies leverage on manipulating the values in various fields (e.g. Unit ID, FC, Data) being transmitted or stored in some way using the ModbusTCP protocol, they all work within the existing structure of the ModbusTCP packet. In the Field Flooding attack presented in this chapter, the ModbusTCP packet structure itself is manipulated, compromising the controller (PLC/RTU), resulting in adverse behaviour outside the intended response as designed.

Intrusion Detection Systems (IDS) for ModbusTCP: Radoglou et al. [166] developed a novel anomaly-based IDS called ARIES which adopted a set of machine learning (ML) methods, consisting of three detection layers: (a) network flow-based detection, (b) packet-based detection, and (c) operational data-based detection. Particularly, the second layer of their model inspects ModbusTCP packets and their attributes to detect anomalies such as unauthorised ModbusTCP commands and function code enumeration attacks. Specifically, they used real datasets originating from a power plant in Greece containing operational data which was used to detect anomalies. Their proposed method is suitable for a specific domain (i.e. power plant) and not for general industrial use-case. Satyanarayana et al. [162] also examined the vulnerability of ModbusTCP to false command injection, false access injection, and replay attacks. Their proposed IDS involved using a frame filtering module that will send only authorized commands and Modbus requests to the PLC by checking the IP address and port of the Modbus client, allowed function codes, and allowed register addresses. Furthermore, Saharkhizan et al. [167] designed an IDS using Deep Learning (DL) long short-term memory (LSTM) modules into an ensemble of detectors which was trained and evaluated on a simulated Modbus network traffic dataset. The dataset was categorised into MitM attacks, ping DDoS (Distributed Denial of Service) flood attacks, Modbus query flood attacks, and TCP SYN DDoS flood attacks which are mostly ”high-rate” attacks. These attacks typically work by sending a series of packets to a target device at a hyper-increased rate, exhausting the capacity for a timely response, if any. The authors focused on detecting mostly ”high-rate” attacks, which can be easier to detect based on the high packet flow. However, they have not evaluated their system against attacks that may be more sophisticated and disguised like the one presented herein (i.e. Field Flooding attack). Therefore, there is no evidence that the proposed IDS could be utilised for detecting such attacks. Also, the attacks used in [167] did not alter the ModbusTCP packet structure.

Finally, the authors in [168] and [169] describe a comprehensive set of rules that could be combined with popular signature-based IDS (e.g. Snort, Suricata) to prevent exploitation of the Modbus protocol. In [169], the authors carried out DoS (SYN Flood), MitM (spoofing), and reconnaissance attacks on a cyber-physical system via ModbusTCP and created custom rules focusing on the Modbus data field, which is plant-specific. A limitation of their work is that these rules would not apply to any other industrial network and is therefore not an adaptable solution. The advantage of an ML-based IDS over this system is its adaptability (ability to learn features of multiple industrial environments) and that it could detect a wider variety of attacks. Both studies - [168] and [169] - examined rules that preserve the integrity of the Modbus packet, but did not consider manipulation attacks where malicious fields are appended to the packet while the parameters within each field remain valid. Also, deploying these rules to adequately protect OT networks requires an in-depth knowledge of various thresholds and set points. Since each OT network has its own unique parameters, thresholds that adequately protect one network may not work as efficiently on another. This solution is not scalable or adaptable across several industrial networks. To summarise, these studies did not consider attacks that abuse the memory allocation of the PLC while preserving the integrity of the Modbus frame. Subsequently, the field flooding attack described in this chapter demonstrates the ability to bypass these preventive techniques by ensuring that the malicious packet is coming from an authorised IP address/port and probing using legitimate function codes and allowed register addresses. Table 4.3 summarises the studies discussed in this sub-section.

4.3 Attacking the ModbusTCP Protocol

4.3.1 Attacker Model and Capabilities

The attacks presented in this chapter consider the following basic assumptions to form the attacker model. OT networks can often include remote access for vendors to maintain their systems remotely. An attacker could perform a phishing attack against a supplier or an integrator/ vendor's remote access link to the OT network [170]. In order to effectively troubleshoot, upgrade or modify system parameters (e.g. PLC logic, proprietary software, hardware configuration files, firmware updates, etc.) during scheduled or emergency maintenance activities, vendors would require administrative privileges on the remote workstations they connect into. This is usually the case, especially in oil and gas offshore platforms located thousands of miles away from shore. It is assumed that our attacker has gained access to the OT network and has the following capabilities: (i) network sniffing; (ii) command injection through scripting; (iii) modification of operational parameters. These capabilities will be further mapped out using the Mitre ATT&CK framework in Section 4.5. The attacker's objectives/ motivation are:

- To compromise an operator's ability to control processes on the remote system (i.e. impair process control).

- Collect information about operational processes including sensor readings and process state.

- Disrupt a process to damage equipment (potentially leading to loss of life and dam-

Table 4.3: Summary of related work. *FF* = *Field Flooding*, Hybrid testbed = ◐

Author / Reference	**Simulation**	**Physical industrial testbed**	**Multiple vendor hardware**	**Alter ModbusTCP packet structure**	**Can detect FF attack**
Chattha et al. [159]	●				
Luswata et al. [160]	●				
Parian et al [161]	●				
Bashendy et al. [163]		●			
Stranahan et al. [164]		●			
Alcaraz et al. [165]	◐				
Radoglou et al. [166]		●			
Satyanarayana et al. [162]		●			
Saharkhizan et al. [167]	◐				
Katakulic et al. [169]		●			
Morris et al. [168]		●			
This study		●	●	●	●

age to the environment).

The tools used in these attacks are:

1. **Smod**: `Smod` is the most widely known pen-testing tool related to ModbusTCP [171]. It aggregates a set of diagnostic and offensive features that can be used in pen-testing the ModbusTCP protocol.

2. **Scapy:** `Scapy` is an interactive packet manipulation program written in Python. It is capable of forging or decoding packets for a wide number of protocols, sending them on the wire, capturing them, matching requests and replies, and much more. [172].

3. **Wireshark:** `Wireshark` is a widely-used network protocol analyzer [173].

4. **Tshark:** `Tshark` is the terminal version of `Wireshark`

5. **Nmap:** A widely used network discovery tool.

4.3.2 Description of Attacks

As discussed in 4.3.1, the attack scenario assumes the attacker has gained entry into the OT network by gaining user credentials of a third-party vendor from spear phishing activities, then using the stolen credentials to access a dedicated workstation with privileges to carry out maintenance activities. The workstation is running an HMI that constantly polls the PLC for process state and displays the status for the operator in real-time. The attacker's targets are highlighted in Figure 4.3

In the initial phase of the attack, the attacker used `Nmap` and `Smod` tools to carry out reconnaissance of the network. `Nmap` was used to discover devices with port 502 (default port used by ModbusTCP protocol) open, while `Smod` was used to scan the PLC for allowed function codes. Due to recent attacks, most vendors no longer allow diagnostic function codes to be sent to PLCs as they can easily be used to discover details about the system, shut it down, or force it into a limited service mode (e.g. Force listen-only mode).

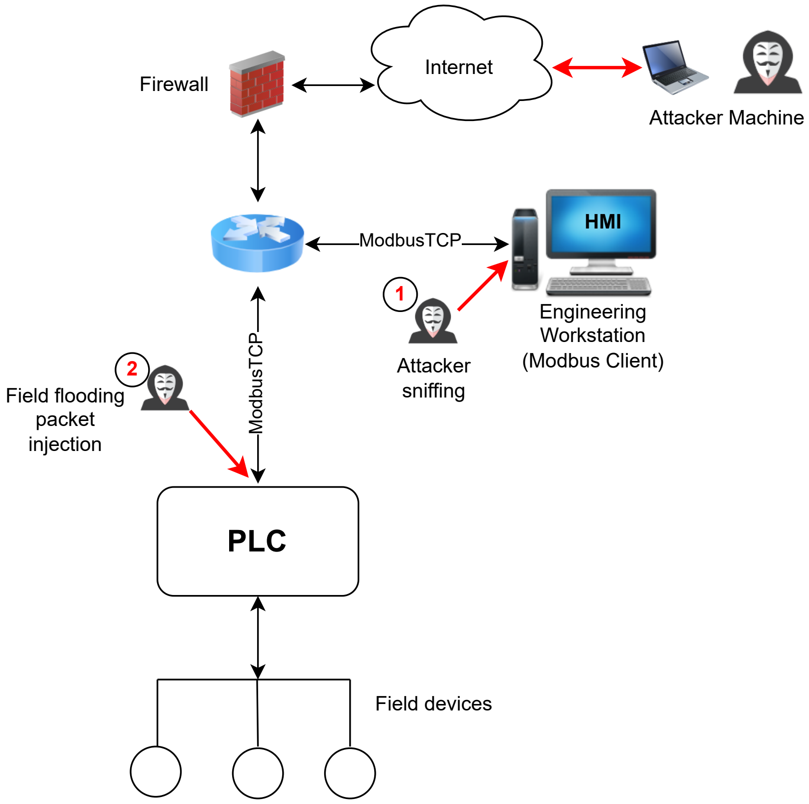


Figure 4.3: Attacker's target points within the OT network; 1 = Field Flooding step 1, 2 = Field Flooding step 2

The `Smod` enumeration on Modbus function codes confirmed that diagnostic function codes are disabled by the vendor on the PLCs. However, all the public FCs shown in Table 4.2 were accessible for exploitation.

The next phase of the attack involved using `Scapy` to sniff network traffic between the HMI and PLC which was analysed with `Wireshark`. ModbusTCP communication between HMI and PLC is usually in a continuous loop. The communication loops in the case of the experimental setups used in this study are described as follows (testbeds are described in detail in 4.3.3):

- **Testbed 1:** one query (to read 2 holding register addresses), its corresponding response (from Modbus server - PLC 1), and finally an acknowledgement (ACK) from HMI - 3 packets.

- **Testbed 2:** two queries (HMI polling PLC for data/status) and two responses (PLC sending requested data/status to HMI). Each query (from HMI) is followed by a corresponding response (from PLC) and an acknowledgement of receipt of data by the PLC - 6 packets.

- **Testbed 3:** one query (to read 1 coil address), its corresponding response (from Modbus server - PLC 3), and an acknowledgement (ACK) from HMI - 3 packets.

In all experiments, the critical metric was the communication time, which was approximately 7ms (milliseconds) between a query-response-ack loop, and 100ms between loops. This gave an initial indication of when malicious packets can be injected into the stream as shown in Figure 4.4. The longer the communication time, the easier it is for `Scapy` to craft a packet and inject. From the `Wireshark` analysis, the time window most favourable for a successful packet injection was the 100ms between PLC acknowledgement for receiving holding register data and HMI requesting input register data in the case of testbed 2. For both testbeds 1 and 3, the packet injection window was after the ACK of the loop, but before the next query from the HMI which also was approximately 100ms. To craft a packet that will be accepted by the PLC, it needs to:

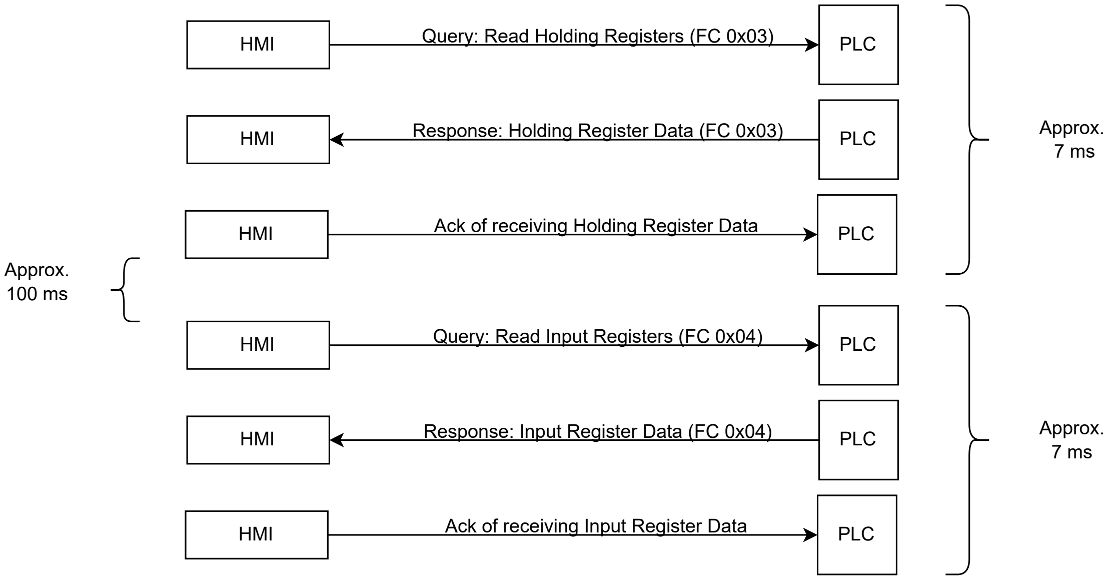

HMI
Query: Read Holding Registers (FC 0x03)
PLC
HMI
Response: Holding Register Data (FC 0x03)
PLC
Approx.
7 ms
HMI
Ack of receiving Holding Register Data
PLC
Approx.
100 ms
HMI
Query: Read Input Registers (FC 0x04)
PLC
HMI
Response: Input Register Data (FC 0x04)
PLC
Approx.
7 ms
HMI
Ack of receiving Input Register Data
PLC

Figure 4.4: PLC-HMI Communication loop showing timings in milli-seconds

- conform with the ModbusTCP standard format (contain function code, transaction and protocol identifiers, unit ID, length and register starting address);

- utilise sequence (SEQ) and ACK numbers in the previous packet (ACK packet transmitted from HMI) to use as its own SEQ and ACK numbers.

Secondly, in order to generate a malicious packet targeting the PLC with a Field Flooding attack, the following techniques were used: (i) alteration of the length field in the MBAP header; (ii) alteration of the number of fields in the PDU header. Recall that the maximum memory allocated for ModbusTCP ADU header is 260bytes. By altering the length field in the MBAP header and increasing the number of fields in the PDU layer, this limit is exceeded which can potentially disrupt the communication between the HMI and PLC. The following experiments were carried out with varying parameters:

- Create ModbusTCP read packet (FC 01/03/04) similar to communication loop packets and inject (packet replay attack).
- Modify ModbusTCP write packet (FC 05/15/06/16) with increased length field in MBAP header and inject (altered length attack).
- Modify ModbusTCP write packet (FC 05/15/06/16) with 1 additional field (2bytes) in PDU layer and inject (Field Flooding attack).
- Modify ModbusTCP write packet (FC 05/15/06/16) with 2 additional fields (4bytes) in PDU layer and inject (Field Flooding attack).

A summary of the Field Flooding attack sequence steps and corresponding stages on the cyber kill chain is shown in Figure 4.5.

4.3.3 Experimental Setup

To carry out these experiments, three testbeds with relevant hardware from real industrial network communications in critical infrastructure were used. The first testbed was purposefully designed and built for this research (described in Chapter 3), while the additional two testbeds were already existing testbeds, built and configured by industry partners.

These three different testbeds were used in order to evaluate and investigate the impact

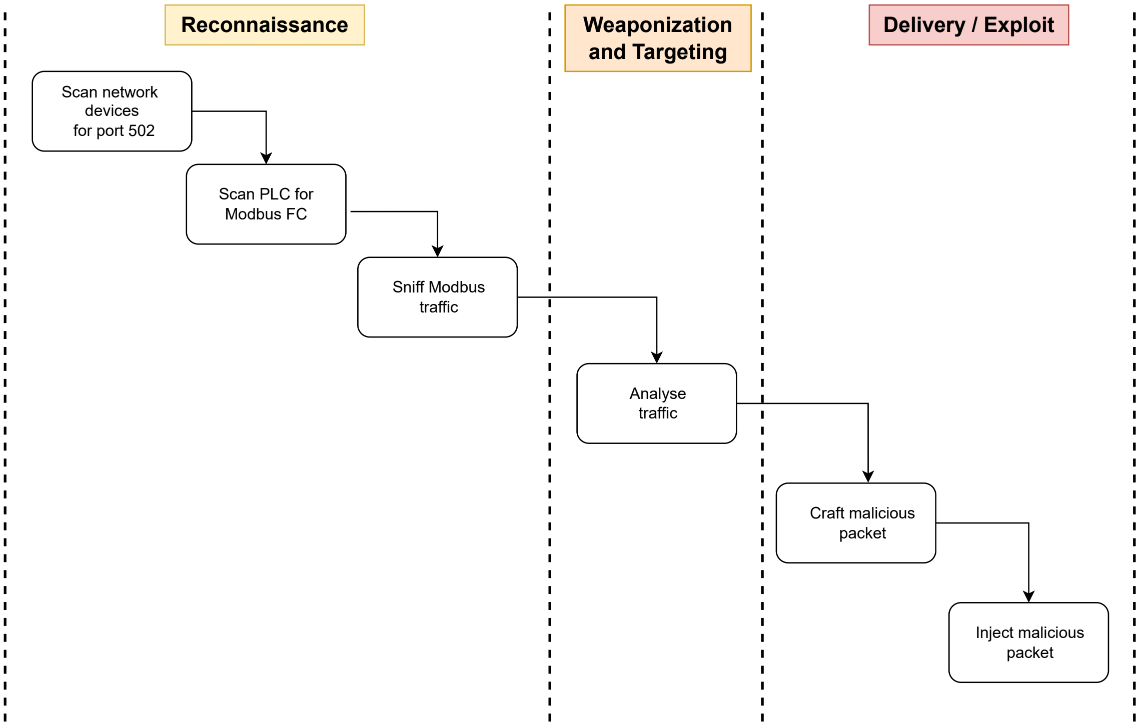


Figure 4.5: Field Flooding attack sequence and phases on the cyber kill chain. FC = Function Code

of the attack on different industrial environments. As each industry vertical has unique operational technologies and communication polling time requirements for safety-critical operations, this allows us to identify how the field flooding attack might propagate and impact their respective operations. The use of multiple testbeds also ensures that the findings are more generalisable and can be applied to a broader range of scenarios, thereby enhancing the validity of our results. An additional reason why we have opted for testbeds with real hardware over simulated testbeds is the opportunity to observe unforseen variables. Real systems often have unexpected behaviours and interactions that are difficult to replicate in simulations.

The main features of the testbeds, such as the PLCs (acting as Modbus servers) and their common industry use cases are listed in Table 4.4. The PLC brands have been concealed for security reasons.

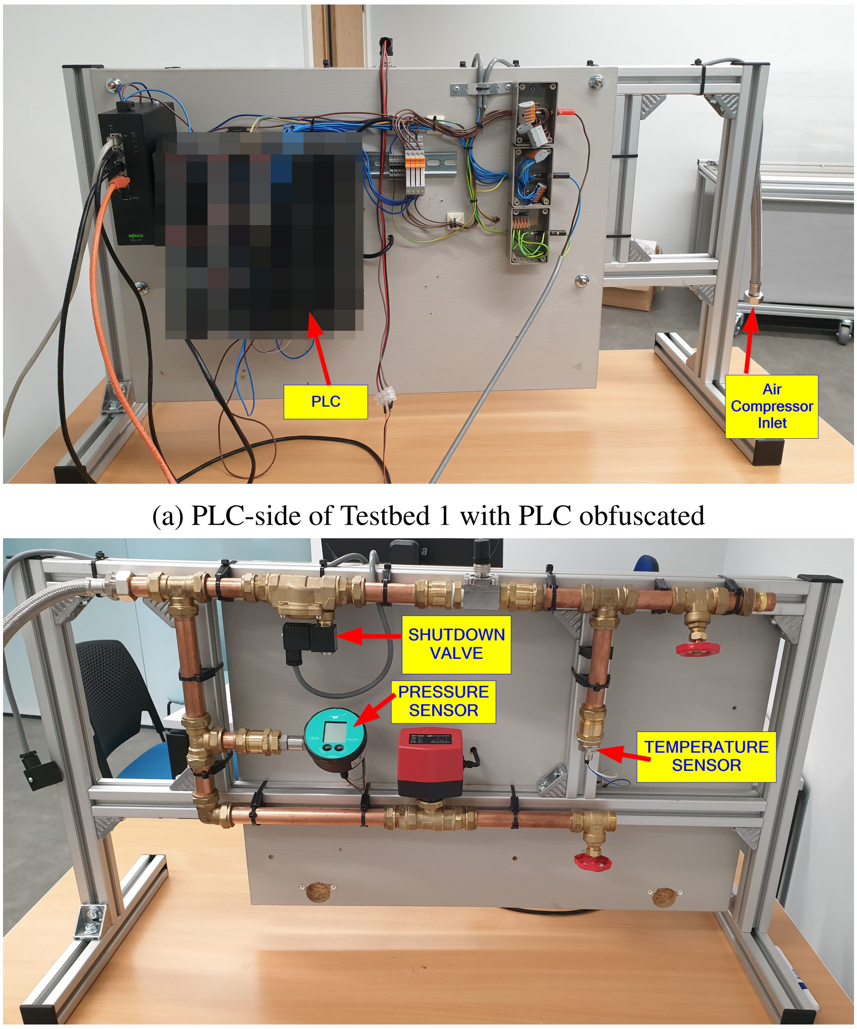
Testbed 1 (Oil and Gas): This testbed emulates a gas wellhead production monitoring system using compressed air flowing through the pipes. The PLC (PLC 1) is commonly

Table 4.4: Common industry use-cases for the 3 PLCs used in our experiments

PLC	Common Industry Use-Case
PLC 1	Oil and gas industry
PLC 2	Manufacturing, smart buildings, general automation
PLC 3	Smart grid, manufacturing

deployed in oil and gas platforms because of its numerous control functions and ability to withstand operations in harsh environments like offshore platforms. An air compressor is connected to the pipe inlet (Figure 4.6a) which pumps compressed air through the system. Monitoring equipment (shown in Figure 4.6b) includes pressure and temperature sensors and a shutdown valve. The values of the sensor readings are stored in the PLC holding register addresses 40099 and 40199. To control the testbed and monitor sensor values, an HMI software, AdvancedHMI [174] provides a Graphical User Interface (GUI) which accesses the stored values in the PLC holding registers and displays the sensor readings (i.e. pressure and temperature). The HMI was programmed to periodically poll the PLC for data representing sensor readings stored in the holding registers using the function code 0x03 (read holding registers). All communication is via ModbusTCP. There is also a shutdown valve to provide the operator ability to shut off airflow emulating an emergency shutdown scenario. This can be controlled via the HMI ”on/off” buttons using FC 0x06 (write holding register).

Testbed 2 (Manufacturing): This testbed represents a simple setup that monitors the temperature and humidity readings of an assembly line to ensure the quality of production. The setup was provided by the National Digital Exploitation Centre (NDEC) and included an encrypted VPN (Virtual Private Network) tunnel to access the testbed remotely. This was to emulate a remote workstation monitoring system process. The hardware comprises a PLC, a temperature sensor, and a humidity sensor. The sensors are hard-wired to the PLC, which communicates the values in real-time to the HMI (Figure 4.7a) using ModbusTCP. For demonstration purposes, both sensors are only reading the temperature and humidity of the room where the testbed is located. These sensor readings are constantly polled and displayed in real-time on the HMI – a feature that allows the operator



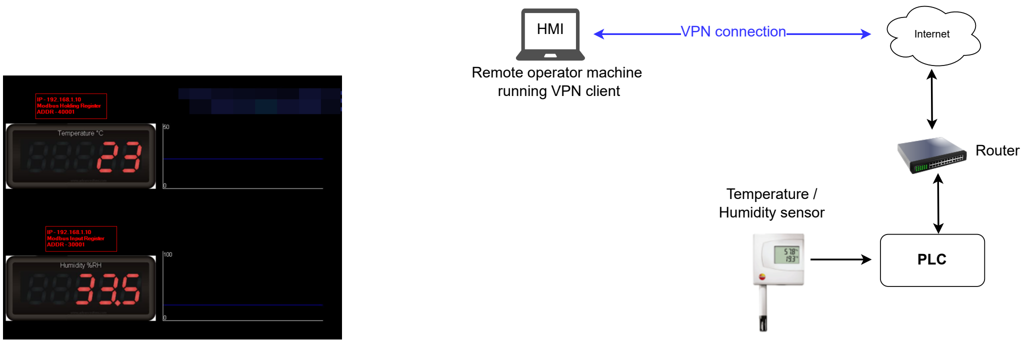
(a) PLC-side of Testbed 1 with PLC obfuscated

(b) Sensor-side of Testbed 1

Figure 4.6: Setup of testbed 1 showing sensors, valves and setup arrangement

to keep track of production quality. The temperature value is stored in a holding register while the humidity value is stored in an input register. The HMI periodically polls the PLC for the temperature and humidity values using the Modbus function codes 0x03 (read holding registers) and 0x04 (read input registers) respectively. The testbed setup is shown in Figure 4.7b.

Testbed 3 (smart city): This is a SCADA testbed consisting of two critical infrastructure systems a) smart city buildings and b) a train system looping around the city. These two systems are controlled separately by two different PLCs. Our study focused on the PLC controlling the smart city buildings. Within the building models (shown in Figure 4.8), there are LED (Light Emitting Diode) lights wired to connect each building to a



(a) HMI used to poll PLC for sensor readings

(b) Setup of Testbed 2

Figure 4.7: Setup of testbed 2 showing remote operator access and HMI used (security details obfuscated)

power source, provided by the PLC. When energised, all the buildings are powered up and illuminated. This is controlled by binary coil values stored in the PLC indicating status as "on" or "off" – indicating when lights in the building can be turned on/off. Auxiliary power lines are included on the surface of the testbed as an aesthetic feature. The HMI tracks the power status of the smart city buildings by accessing the values stored at coil address 0001 using FC 0x01 (read coil) and gives the operator the ability to turn on the power, or power down (using FC 0x05 - write single coil) for maintenance activities.

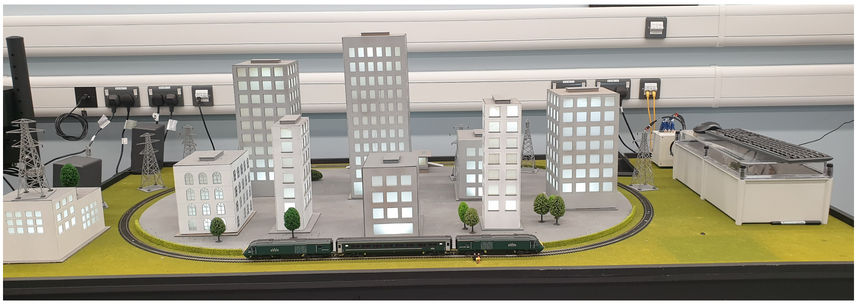


Figure 4.8: Complete setup of Testbed 3

In the next section, the results of these attacks on all three testbeds are described.

4.4 Results

Malicious packets were successfully injected into the ModbusTCP communication for all the testbeds with each packet altered according to the experiments listed in 4.3.2. In each testbed, sniffing network traffic and injecting the exact same ModbusTCP read packets in the communication loop resulted in the corruption of the TCP session, however, the TCP protocol session management was able to self-correct with minimal disruption - approximately 1 second (i.e. spurious retransmissions were discovered and the TCP three-way handshake was re-initiated to re-establish communications).

The next type of malicious packets that were injected was the altered length field in the MBAP header. Again, for all three PLCs in the various testbeds, the results were similar. The injected packet with an increased length field corrupted the TCP session and the session management self-corrected the communication. However, in this case, the communication loop was restored after a RST ACK packet which triggered the re-initiation of the TCP three-way handshake. This also took approximately 1 second to correct and all 3 PLCs handled this error adequately. It's also worthy to note that rule No. 3 in [168] will effectively block this attack. The aim of this experiment was to establish a baseline for the PLCs' error handling capabilities.

Finally, malicious packets with additional fields (field flooding attack) to the PDU header were injected and all three PLCs behaved differently in handling this attack. Each malicious Field Flooding Packet injected (disguised as a Modbus client query) triggered an initial response to the sent query from all three PLCs, which confirmed a successful packet injection and enabled a continuation of the attack (maximum of 4 packets injected) until the PLC is unable to respond to further legitimate requests for varying periods. The impact on each testbed is further described as follows:

Field Flood Attack on Testbed 1: Two types of malicious ModbusTCP packets with additional fields in the PDU header were injected to cause a field flood attack on PLC 1. The first packet was injected with only 1 additional field (2 bytes) while the second packet had 2 additional fields (4 bytes). The first packet (additional 2 bytes) caused a denial of service for up to 5 minutes where the PLC (modbus server) did not respond to queries from the HMI (modbus client). The second field flood attack (2 additional fields - 4 bytes) had a more damaging impact on the modbus server as the PLC was continuously responding to queries from HMI with RST ACK packets in an attempt to reset the TCP session. The field flooding attack effectively forced the PLC into a listen-only mode for approximately 59 minutes leading to a denial of service. This is shown in Figure 4.9.

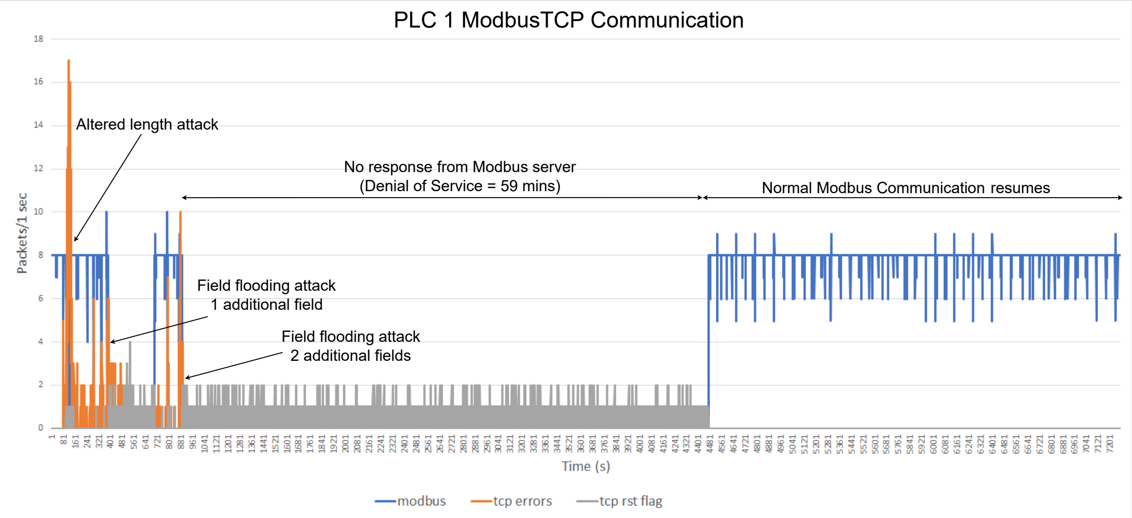


Figure 4.9: Disruption of ModbusTCP communication from field flooding attack on Testbed 1

Field Flood Attack on Testbed 2: The field flooding attack also showed adverse behaviour on PLC 2. Although the injected packet with only 1 additional field in the PDU header resulted in a corruption of the TCP session for 9 seconds, when repeated with a malicious field flooding packet containing 2 additional fields, it resulted in a denial of service. The additional 4 bytes appended to the PDU header made the PLC non-responsive to HMI queries by sending RST ACK packets for approximately 7 minutes.

Field Flood Attack on Testbed 3: For PLC 3 (smart city testbed), the field flooding attack was also carried out by injecting malicious ModbusTCP packets with 1 additional field and 2 additional fields. The field flood attack with 1 additional field to the PDU header corrupted the TCP session for about 20 seconds, while that of 2 additional fields forced the PLC to restart as shown in Figure 4.11. This also caused a denial of service scenario as, during the period of the restart, the PLC would no longer be responsive to commands or report process state.

The summary of all the attacks carried out on the testbeds and their corresponding impact on the behaviour of the PLCs is shown in Table 4.5.

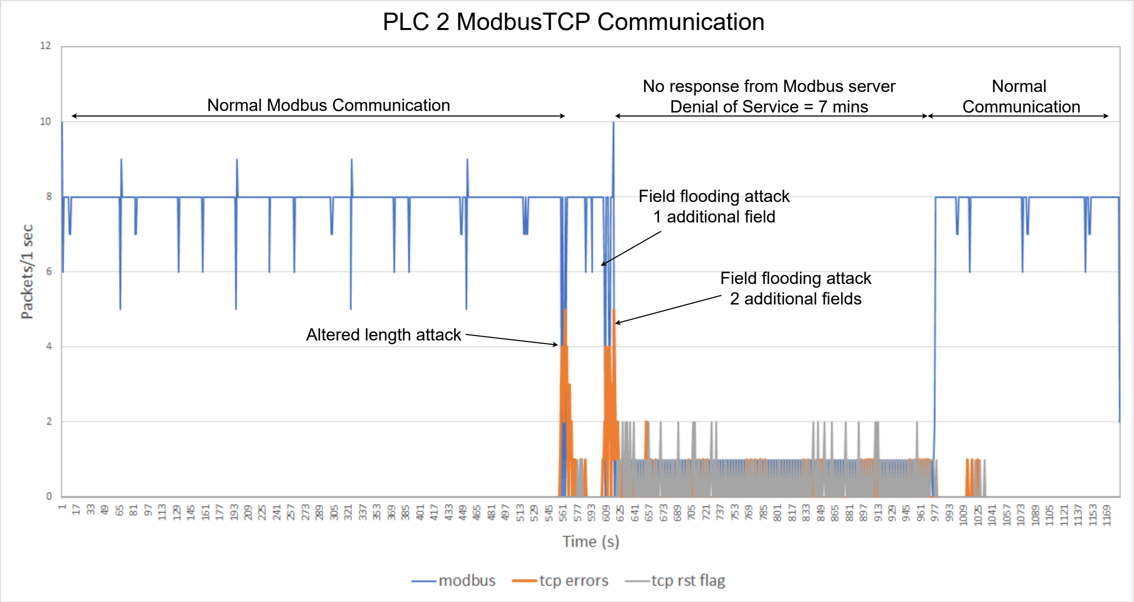


Figure 4.10: Disruption of ModbusTCP communication from field flooding attack on Testbed 2

4.5 Analysis of Field Flooding Attack Impact

From the results shown in Section 4.4, it can be deduced that different PLCs behave uniquely to the field flooding attack. This is a unique advantage that real systems (i.e. physical testbeds) have over simulated environments as this difference in PLC behaviour cannot be accounted for in simulated experiments. Our experiments show that PLC 1, which is predominantly used in the oil and gas industry, is the most vulnerable to the field flooding attack in comparison to PLCs 2 and 3. This could potentially have serious implications on process safety in such a volatile, critical industry. For example, in oil and gas production platforms, where SCADA is used to control the heating and separation of volatile hydrocarbons, operators monitor and ensure safe operations via HMI equipped with override functions for emergency shutdowns. This attack has the potential to impair process control leading to pipeline explosions, loss of lives and damage to the environment.

One of the dangers of the field flooding attack is that a low-skilled adversary can execute this attack and cause huge damage. Its relative ease of execution can be demonstrated by mapping the attack pattern on the Mitre ATT&CK for ICS framework. This framework

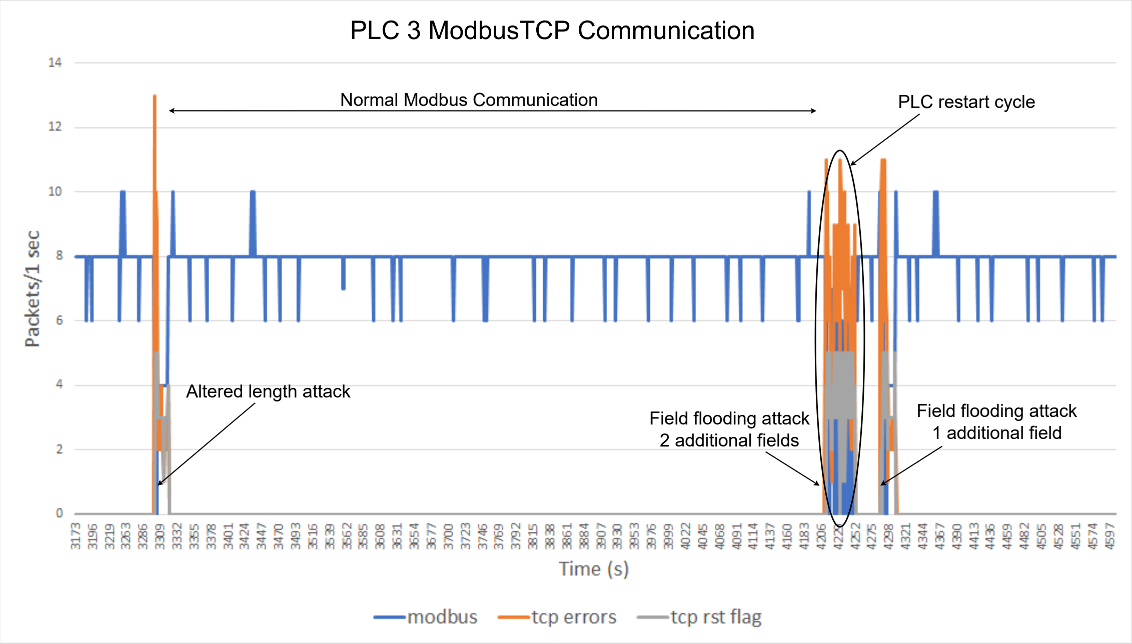


Figure 4.11: Disruption of ModbusTCP communication from field flooding attack on Testbed 3

is a curated knowledge base for cyber adversary behavior in the ICS technology domain [175]. It comprises a taxonomy that describes adversarial tactics and techniques.

Using the Mitre ATT&CK for ICS framework the field flooding attack was mapped to show the tactics and techniques utilised by the attacker. Out of 12 available tactics, only 6 were required to achieve the attacker's goal of Denial of Control (T0813) and Denial of View (T0815). The fewer tactics used to reach the desired impact goal, the easier it is to carry out an attack on live production systems. This is summarised in Table 4.6.

4.6 Detection of Field Flooding Attack: Supervised Machine Learning

4.6.1 Dataset

The dataset was created by collecting a combined 4 hours worth of network pcap traffic from all three testbeds using `Wireshark`. During the capture, the PLCs had malicious packets injected into the stream as described in 4.3.2 and the data was saved into three separate pcap files (i.e. one from each testbed). These pcap files were converted into a csv file format using `Tshark`, and subsequently combined into a single file to make a total

Table 4.5: Summary of impact of attacks carried out on all three testbeds

		Attack Impact		
Testbed	**PLC/RTU**	**Altered length attack**	**Field Flooding Attack (1 field)**	**Field Flooding Attack (2 fields)**
1	PLC 1	Spurious retransmissions (1 sec)	Denial of service (5 mins)	Denial of service (59 mins)
2	PLC 2	Spurious retransmissions (1 sec)	TCP session corruption (9 secs)	Denial of service (7 mins)
3	PLC 3	Spurious retransmissions (1 sec)	TCP session corruption (20 secs)	PLC forced restart

Table 4.6: Summary of Mitre ATT&CK tactics and techniques used in field flooding attack

Tactic	**Technique**	**Technique ID**
Initial Access	Internet accessible device	T0883
Execution	Command-line interface, scripting	T0807, T0853
Discovery	Network Sniffing	T0842
Inhibit Response Function	Block reporting message, denial of service	T0804, T0814
Impair Process Control	Modify parameter, unauthorised command message	T0836, T0855
Impact	Denial of control, denial of view	T0813, T0815

of 127,758 data points containing 29 features (114,700 = benign and 13,058 = malicious) – comparable in size to datasets used in other similar studies (e.g. [176], [119]). To label the dataset, it was ensured that every malicious packet injected successfully had the same transaction ID (e.g. 8000). By filtering the field `mbtcp.trans_id` == 8000, the start of each field flooding attack was identified and labelled appropriately to capture its impact. Combining the datasets from the 3 testbeds enabled the development of a more robust model that would generalise better when using data from similar ICS networks. The total attack duration of the experiments carried out was approximately 60 seconds and a summary of the dataset description is shown in Tables 4.7 and 4.8.

Table 4.7: Summary of dataset

Total data points	127,758
Benign data points	114,700
Attack data points	13,058
Total capture duration	3.8 hours

Table 4.8: Summary of attacks in dataset (AL = Altered length, FF = Field Flooding)

Attack Type	**Packets Injected**	**Attack duration (sec)**
Packet replay	3	4.6
AL Injection	3	6.7
FF + 1 Field	6	13.1
FF + 2 Fields	12	35.7
Total	**24**	**60.1**

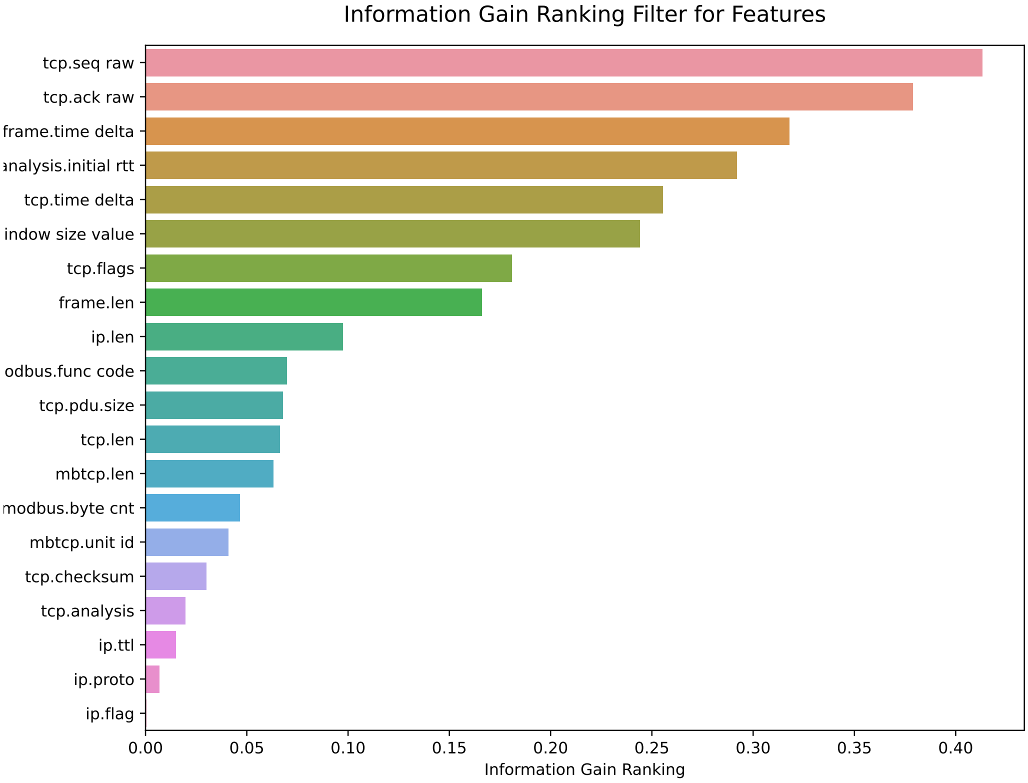
4.6.2 Feature Selection

To train a supervised machine learning model effectively, it is important to identify features that best describe the dataset [177]. As the focus of our study is the ModbusTCP protocol, features from the TCP/IP layers and the Modbus layer (embedded within the TCP layer) form the key selected features for our model training. Features from the ethernet layer (e.g. mac addresses, src and dst addresses) were not considered because they include properties which may lead to overfitting of the machine learning model. At the same time, temporal features from the Frame header (e.g. `frame.time_delta`) to capture packet inter-arrival times were also selected. This created an initial dataset with 30 features with the labelled target variable inclusive.

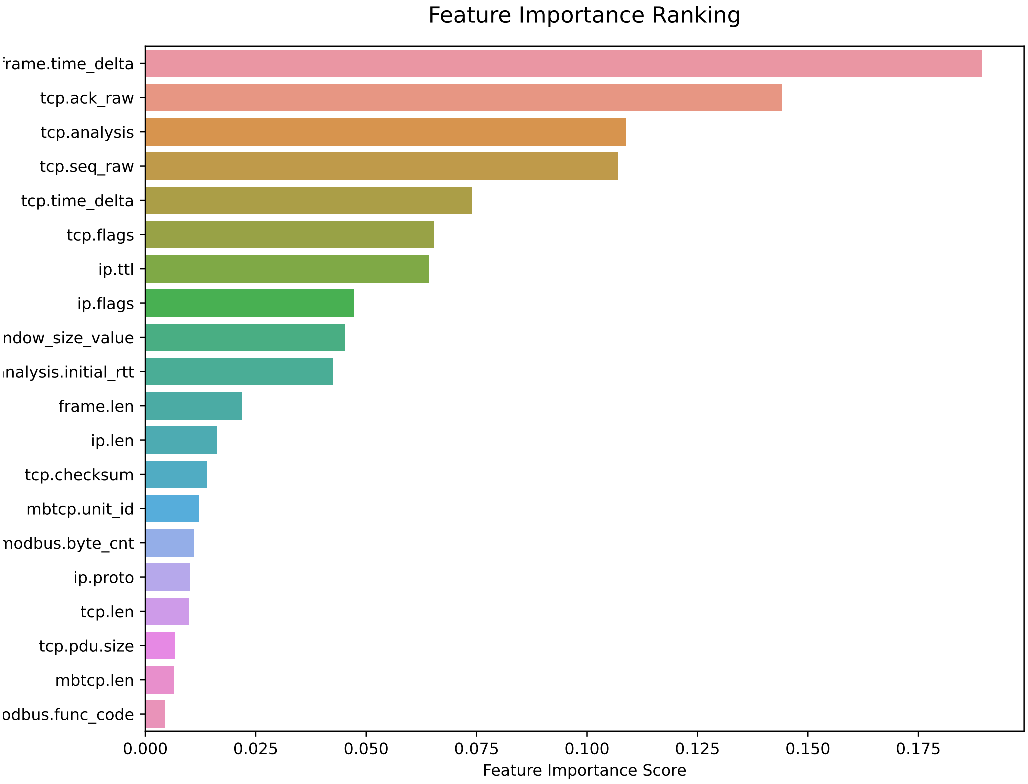
To further reduce the risk of overfitting, features that represent identifying properties (e.g. IP/mac addresses) were also removed from the feature set [177]. In this case, the `mbtcp.trans_id` feature was also removed as all the attacks had the same transaction ID. Furthermore, features that had only one unique value within the dataset were not considered as these would have no effect on the target variable and would increase computational overhead. This resulted in pruning the number of selected features to 20.

Additionally, to understand the worth of each feature for the target variable, two feature selection filters – InfoGainAttributeEval and feature_importances_ – using `Weka` [178] and the scikit-learn library respectively were applied to the remaining 20 features. The InfoGainAttributeEval evaluates the *worth* of an attribute by measuring information gain with respect to the class [179] [177] while feature_importances_ is computed based on how often the feature is used for splitting nodes across all the trees in the ensemble. This identifies features more significant for detecting an attack.

The result of the filters, shown in Figures 4.12a and 4.12b indicate that the delta time, raw SEQ, and ACK attributes are consistently ranked high. More importantly, the delta time attribute in both charts ranks significantly high, which is an indication that the time difference between consecutive packets is a very relevant feature in detecting anomalies in an industrial network. This is critical because the timing of packets (packet inter-arrival times) is a key parameter used to characterise the periodicity of an industrial network. A further possible explanation for the high delta time ranking may be due to the fact that the Field Flooding attack exploited the gap of 100ms between loops in the ModbusTCP transmission to inject the malicious payload, which would invariably lead to distortion of the regular benign delta time packet transmission. Furthermore, the importance of raw SEQ and ACK scores could be attributed to the way the field flooding attack is executed as it measures SEQ and ACK numbers and uses them as the seed to generate a malicious packet.



(a) Information Gain Ranking Filter for features (Weka)



(b) Feature Importance Ranking Score (scikit-learn)

Figure 4.12: Ranking of most important features using InfoGainAttributeEval and feature_importances_

4.6.3 Model Training and Analysis

All machine learning experiments were carried out on a Windows 10 PC with Intel(R) Core(TM) i7-8665U CPU at 1.90GHz processor and 16gb RAM. The final dataset with 20 features selected as discussed in Sec. 4.6.2 went through data pre-processing (i.e. data normalisation and label encoding) before model training. The dataset was randomly split into 60% for training and 40% for testing and evaluation on unseen data. The choice of an appropriate algorithm is based on model performance for a particular problem and the properties of data that characterise the problem [177]. Eight classifiers were considered based on other relevant work [180], [181]; and based on how they operate. In more detail, the models included algorithms that function based on conditional dependencies in the dataset or assume conditional independence (e.g., Bayesian Network and naive Bayes), discriminative models that aim to maximize information gain without modeling any underlying probability or structure of the data (e.g., J48 decision tree and support vector machine), and ensemble models that utilise multiple ML algorithms to produce higher predictive performance than could be obtained from a single ML classifier [179], [182] (e.g. Random Forest, XGBoost).

Before discussing the metrics to be used in evaluating the classifiers, the following terms shall be explained:

- True Positives (TP): Number of actual positives correctly predicted.
- True Negative (TN): Number of actual negatives correctly predicted.
- False Positive (FP): Number of actual negatives predicted incorrectly as positive.
- False Negative (FN): Number of actual positives predicted incorrectly as negative.

In evaluating the performance of our classifiers, it is recommended to use precision, recall, and F1-scores [183] defined as:

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

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

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

The best performing classifier was XGBoost with an F1-score of 99.9% while the joint second best performing classifiers were Random Forest and Decision tree with F1-scores of 99.8% each – making all three top performing classifiers tree-based algorithms. Furthermore, both XGBoost and Random Forest are ensemble algorithms that use decision trees as their meta-classifier and generally perform well on non-linear problems as in our case. Table 4.9 shows the precision, recall, and F1-scores of all evaluated classifiers. The confusion matrices of both XGBoost and Random Forest reveal that the XGBoost classifier predicted marginally less FN/FP than the Random Forest classifier as shown in Table 4.10.

Table 4.9: Classification metrics results

Classifier	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.952	0.992	0.972
Random Forest	0.998	0.998	0.998
Naïve Bayes	0.978	0.676	0.799
Decision Tree	0.998	0.998	0.998
XGBoost	**0.999**	**0.999**	**0.999**
K-NN	0.997	0.997	0.997
Kernel SVM	0.995	0.993	0.994
SVM	0.975	0.973	0.974

Table 4.10: Confusion matrices for XGBoost and Random Forest classifiers

		Predicted	
		Malicious	Benign
Actual	Malicious	5,189	65
	Benign	49	45,801

(a) XGBoost

		Predicted	
		Malicious	Benign
Actual	Malicious	5,181	73
	Benign	75	45,775

(b) Random Forest

4.7 Conclusions

With the increase in cyber attacks on Industrial Control Systems and the frequency of those attacks leading to DoS scenarios, this chapter identifies a pathway to attacking these systems to deny legitimate service using the ModbusTCP protocol. Previous work has focused on protecting ModbusTCP packets by ensuring the size allocated to a particular field in the MBAP and PDU headers are within set limits. In this research, a novel field flooding attack capable of bypassing these protection mechanisms was demonstrated, keeping the fields within their data size (in bytes) limit, but increasing the number of fields by 2, resulting in an additional 4 bytes of fields to the PDU header.

The impact of the field flooding attack was evaluated on three physical industrial testbeds with different configurations. The results show that the PLC usually deployed in the oil and gas (OG) industry was the most vulnerable to this attack as one malicious packet resulted in a denial of service of approximately 59 minutes. In OG operations this could have significant implications as it could potentially lead to unsafe conditions which could damage the environment due to the hazardous nature of hydrocarbons. Although this attack has been shown to be capable of disrupting OG operations, it could also potentially disrupt critical ICS communications in other sectors. This work also shows that PLCs may behave differently to the same cyber attack, which highlights a clear advantage of using real industrial testbeds for security research and the limitations of simulated cyber-physical testbeds – as these simulated experiments are unable to account for the difference in PLC behaviour in a real system.

To effectively detect the Field Flooding attack, our initial machine learning experiments demonstrated that the best performing classifier was XGBoost – an ensemble algorithm based on a Decision Tree meta-classifier. This chapter presented the initial experiments for automatically detecting attacks using machine learning algorithms by utilising signatures from pcap files. The positive findings indicate that supervised machine learning can be used to effectively detect cyber attacks in an industrial network environment. This is evidenced by the high F1-scores achieved by the selected classifiers. However, to successfully deploy a supervised ML-based IDS in this sort of environment, there must be access to a labelled dataset to train the model – which is usually a difficult task given the huge costs and time required to acquire or label the dataset(s) respectively. Another limitation of this method is its inability to detect zero-day attacks.

A significant finding in this chapter was the ranking of the delta time attribute in the feature selection process. The consistently high ranking score attained by the delta time feature (packet inter-arrival times) indicates that it is a very relevant feature in detecting anomalies in an industrial network. This is critical because it is key parameter used in determining the periodicity of an industrial network.

Subsequent chapters of this thesis attempt to overcome the identified limitations by investigating unsupervised methods of anomaly detection. First, unsupervised ML classifiers are evaluated (Chapter 5) which leads to the development of a novel unsupervised method that works with the principle of periodicity in industrial networks (Chapter 6). Finally, the adaptability of the model developed in Chapter 6 is evaluated on a public dataset to determine its effectiveness in different industrial environments in Chapter 7.

Chapter 5

Detection of attacks on industrial networks using unsupervised machine learning

Parts of this chapter are included in the paper "STADe: An Unsupervised Time-Windows Method of Detecting Anomalies in Oil and Gas Industrial Cyber-Physical Systems (ICPS) Networks" currently under review

5.1 Introduction

In chapter 4, we evaluated the performance of eight supervised machine learning classifiers to determine their effectiveness in detecting the field flooding attack on ICPS. The experiments showed promising results with the tree-based algorithms (i.e. XGBoost, random forest, and decision trees) having the best results. Overall, all the classifiers had above-average performances considering their high F1 scores. This shows that with access to a labelled dataset, supervised machine learning can be utilised to design an effective intrusion detection system for ICPS. However, in reality, most do not have access to a labelled dataset considering the high costs and substantially time-consuming effort required to produce them [184]. This makes unsupervised methods of anomaly detection a more attractive prospect as it does not have this requirement and also has the added benefit of potentially detecting zero-day (unknown) attacks. However, despite these significant advantages, unsupervised machine learning methods are yet to be adopted significantly in industry because of the main challenge of high false alerts [113]. This limitation would need to be reduced significantly to improve the chances of adoption in real operational environments.

This chapter introduces the initial anomaly detection experiments using unsupervised machine learning algorithms to detect cyber attacks against ICPS. The aim is investigate how effective the methods are at detecting anomalies, and more importantly, investigate which of the methods records the least false positives. This could form the basis for further studies in improving the technology. If found to be effective, unsupervised machine learning methods have the potential to detect zero-day attacks (never before seen attacks) in contrast to supervised methods which are signature-based. This is what makes unsupervised methods of anomaly detection very attractive for further investigation, and thus, forms the motivation behind the fourth research question:

RQ4 *How efficiently can unsupervised machine learning methods be utilized to detect anomalies in industrial cyber-physical systems?*

In answering this question, this chapter will provide the fifth and sixth contributions of this research:

C5 *An investigation into how unsupervised machine learning algorithms can be utilised for anomaly detection in industrial cyber-physical systems.*

C6 *This research contributes a catalogue of labelled industrial network datasets in csv format including the original pcap files containing benign and attack data. The attacks carried out are field flooding attacks, SYN flooding attacks, and Man-in-the-Middle attacks.*

5.2 Utilising Unsupervised ML Models to Detect Anomalies

IDS systems based on unsupervised machine learning models are increasing in popularity as most of the available data for research is unlabelled. The central idea is to learn a model of normality from normal data in an unsupervised manner so that anomalies become detectable through deviations from the model [185]. In essence, anomalies are expected to exhibit different characteristics when compared to other points. A number of algorithms have been proposed in the literature for anomaly detection like the graphical method, statistic method, distance-based method, density-based method, and model-based method [186]. Of these, the most frequently used are the distance-based method (e.g. K-Nearest Neighbour, KNN), density-based method (e.g. Local Outlier Factor, LOF), and the model-based method (e.g. isolation forest). This is mainly because of their ability to detect global and local (deeper lying) outliers especially when mapping high-dimensional data unto a low-dimensional subspace (as in the case of KNN and LOF) and also explicitly isolating anomalies rather than profiling normal instances (as in the case of isolation forest). As a result, these models perform well with high dimensional data with a low memory requirement. Isolation forest, being a tree-based algorithm, is of particular interest in this case because of the performance of such models in the experiments carried out in chapter 4. For these reasons, this chapter will focus on investigating, adapting, and evaluating KNN, isolation forest, and LOF algorithms for the purposes of i) detecting the novel field flooding attack, and ii) detecting attacks earlier identified in Chapter 2 as impacting oil and gas systems. The aim is to investigate and understand how well unsupervised ML models can identify low-rate attacks (i.e. field flooding attacks and MITM) combined with high-rate attacks (i.e. SYN flooding attacks).

5.2.1 KNN Algorithm

The distance-based method is generally regarded as the basic method of anomaly detection research [187]. Ramaswamy et al. [188] proposed the anomaly detection algorithm KNN as a distance-based algorithm that ranks each point based on the distance between the point and the nearest point, and identifies the top-most points as anomalies. A distance function - typically the Euclidean distance function - is required to determine the distance between the data and their k^{th} nearest neighbour to quantify the degree of the anomaly in an unsupervised manner. To determine if a data point is an outlier, its number of neighbours within a radius r must be less than k [189]. The authors in [190] used KNN to analyse and detect anomalies in a wireless sensor network. Their proposed method was tested on the QualNet simulation platform and proved to detect anomalies with relatively low errors. However, a key limitation in their approach is that they were based on simulated experiments which the authors admit must be combined with real-life scenarios. The authors in [191] utilise KNN alongside other algorithms in a ML-based anomaly detection model to identify attacks on a power grid Distributed Control System (DCS) by monitoring the network traffic data. They extracted 9 statistical features including payload data, number of packets, and destination addresses. Similarly, Phillips et al. [192] evaluated anomaly detection models using KNN to classify normal and abnormal instances in SCADA network traffic. However, there were no details regarding the implementation of the algorithms nor the feature selection methods employed. The features used in the model includes source and destination addresses and payload command data. The limitation of these approaches mainly lies in the fact that the model would only apply to the specific operational network it was trained on and thus, is not adaptable to other environments because payload data and source or destination address are features of network traffic that are unique to each industry vertical or critical infrastructure operational environment.

The methods utilised in our approach, similar to Chapter 4 involves removing all network identifying features like source or destination addresses, source or destination ports, and payload data. This would ensure that the risk of model overfitting to the data is minimised, as well as improving model adaptability to other industrial networks.

5.2.2 Isolation Forest Algorithm

The Isolation Forest, first proposed by [193], represents an anomaly detection variant of the popular classification method Random Forest [194]. It relies on the concept that anomalies are few and different from the rest of the data and as such, are more susceptible to isolation. This isolation process involves the repeated partitioning of instances recursively until all instances are isolated, producing shorter paths for anomalies in a tree structure. Isolation forest is suited for network traffic anomaly detection because of its ability to handle high-dimensional data. Zhang et al. [195] utilised Isolation Forest, together with a Locality-Sensitive Hashing (LSH) technique to create an anomaly detection algorithm which was tested on both synthetic and real-world datasets. Their experimental results showed good results, however, it only works well for static data. Industrial network data is a high-dimensional and dynamic data that varies with time. The authors in [196] also proposed an approach involving the use of Isolation Forest algorithm to detect anomalies in a real-time data stream called ASTREAM. The KDDCUP99 dataset was used to verify their method and proved to have high scalability. However, by rebuilding their model repeatedly to update the latest data points, the computational overhead of their method is very high. Aboah et al. [197] used Isolation Forest to detect anomalous behaviour in a traffic light system by tracking the operations at the PLC input and output memory addresses. Results indicated that the Isolation Forest algorithm achieved good model performance, however, their work focused on data collected from PLC input and memory addresses which, when compared to industrial network data, is not high-dimensional as is obtainable in real-world operational environments.

5.2.3 LOF Algorithm

Early outlier detection algorithms were originally focused on detecting global outliers, which were based on finding data points that fall outside the normal range for an entire dataset [198]. However, this can be computationally inefficient because of complexity and data variability. On the other hand, because we mostly care about the change of data in a local scope in many anomaly detection scenarios, local outliers are able to identify anomalies that may fall within the normal range for the entire dataset, but outside the normal range for the surrounding data points [199, 198]. The LOF algorithm was first developed by [200] for the outlier detection problem of KDD applications and has become one of the popular density-based algorithms.

For each data point, the process of identifying the local outlier factor is by calculating the degree of outlying [199] – which depends on the local density of every point k nearest neighbour [201]. The primary method of density-based anomaly detection is to first define the density of the object according to two parameters [198]:

- distance between the objects, and
- number of objects within a given range.

As mentioned earlier, a lot of studies have utilised LOF for anomaly detection, however, not many have applied it to industrial network data which consists of a continuous data stream. Mutua et al. [202] utilised LOF to create a statistical model for anomaly detection in a smart grid. They observed selected features from an ICS communication network for use in building the model, and were able to detect anomalies in an unsupervised manner. Also, the authors in [203] developed hybrid deep learning models by combining deep learning models with LOF to detect cyber-attacks using smart grid datasets. Their results indicated that hybrid models combined with LOF offered a superior detection performance compared to other models in the study. Similarly, Grammatikis et al. [204] evaluated three outlier detection algorithms namely a) LOF, b) isolation forest, and c) One Class SVM (OCSVM) to detect anomalous flows from unauthorised IEC-104 commands and MitM attacks in an industrial environment. Their proposed IDS consisted of two main components – a sensor (responsible for monitoring and analysing the entire network traffic generated) and server (a centralised point where anomaly detection takes place) – which combined to detect anomalous scenarios. However, these studies focused on the IEC 60870-5-104 protocol (also known as IEC 104) which is focused on the power systems and therefore, is unlikely to be adaptable to the oil and gas industry without model retraining on relevant datasets.

To the best of our knowledge, this is the first study investigating unsupervised machine learning models to detect attacks such as the novel field flooding attack (identified in Chapter 4) in industrial networks without considering payload-based features. The data used for the study is collected from normal operations of a gas wellhead monitoring testbed and attack scenarios based on popular attacks identified in chapter 2 that have been carried out against upstream oil and gas assets. In the next section, a detailed description of the dataset, attack scenarios, data pre-processing and evaluation criteria used will be presented.

5.3 Experiments and Dataset Description

The experiments carried out in this chapter were kept as close as reasonably practicable to those in chapter 4. The same 20 features used in the supervised ML IDS in chapter 4 were used for this set of experiments (i.e. after removing features that represent identifying properties like mac/ip addresses) giving a set of 20 features.

5.3.1 Attack Scenarios

The attacks considered for these experiments were derived from the most frequent attacks on oil and gas critical infrastructure targeting ”Availability” and ”Integrity” as described in chapter 2. For attacks targeting availability, field flooding attacks and SYN flooding attacks were carried out while for attacks targeting integrity, a Man-in-the-Middle (MITM) attack was executed. The testbed setup, attacker model, and mode of execution are all the same as were described in chapter 4.

5.3.2 Data Collection

For these experiments, a combined total of 7 hours of data was collected from the wellhead monitoring testbed described in chapter 4 and labelled to enable proper evaluation of the performance of the selected models. This contained instances of each of the 3 attacks under consideration (i.e. field flooding, SYN flooding, and MitM attacks) as summarised in 5.1. A field flooding attack was executed which caused a denial of service to the

Table 5.1: Summary of dataset

Attack	Attack type	Attack Duration	Total Capture Duration	No. of Pkts
Field flooding	DoS	1 hr	7.3 hrs	1,023,202
SYN flooding	DDoS	13 secs		
MitM	Spoofing	5.5 mins		

PLC for a period of 1 hour while a SYN flooding attack (crafted as a DDoS attack from multiple IP addresses) was executed for 13 seconds. Finally, the last attack carried out was a Man-in-the-Middle attack for a duration of 5.5 minutes.

5.3.3 Data Pre-Processing

One core difference in the data pre-processing approach adopted in this chapter is that there is no requirement for a train/test split. This is because anomaly detection works on the assumption that anomalous events are very rare, which in turn, produces highly imbalanced training datasets. As a result, the goal is to learn a valid model of the majority of data points (normal data) [185] which helps it detect deviations from the norm.

Furthermore, dimensionality reduction was applied to the dataset for the KNN and LOC experiments. KNN and LOC perform optimally when high-dimensional data is reduced and projected onto a lower-dimensional space. Therefore, Principal Component Analysis (PCA) was applied to reduce the 20 selected features to a 2-dimensional array. This helps reduce the computational complexity required for detection. PCA is the most common dimensionality reduction technique [205]. By identifying directions of the highest variance from higher-dimensional data and projecting them onto a lower-dimensional subspace, when used with an ML model, it is able to reduce the number of parameters fed into the model without sacrificing important details in the data [206]. Default hyperparameters were retained in most cases except for the following empirically determined optimal choices:

- **KNN:** All default hyperparameter values
- **Isolation Forest:** n_estimators=50, contamination=0.048
- **LOF:** contamination: 0.048

5.3.4 Feature Importance

Similar to Chapter 4, the same feature selection filters were applied to the expanded dataset to determine the worth of each attribute. This is shown in Figure 5.1.

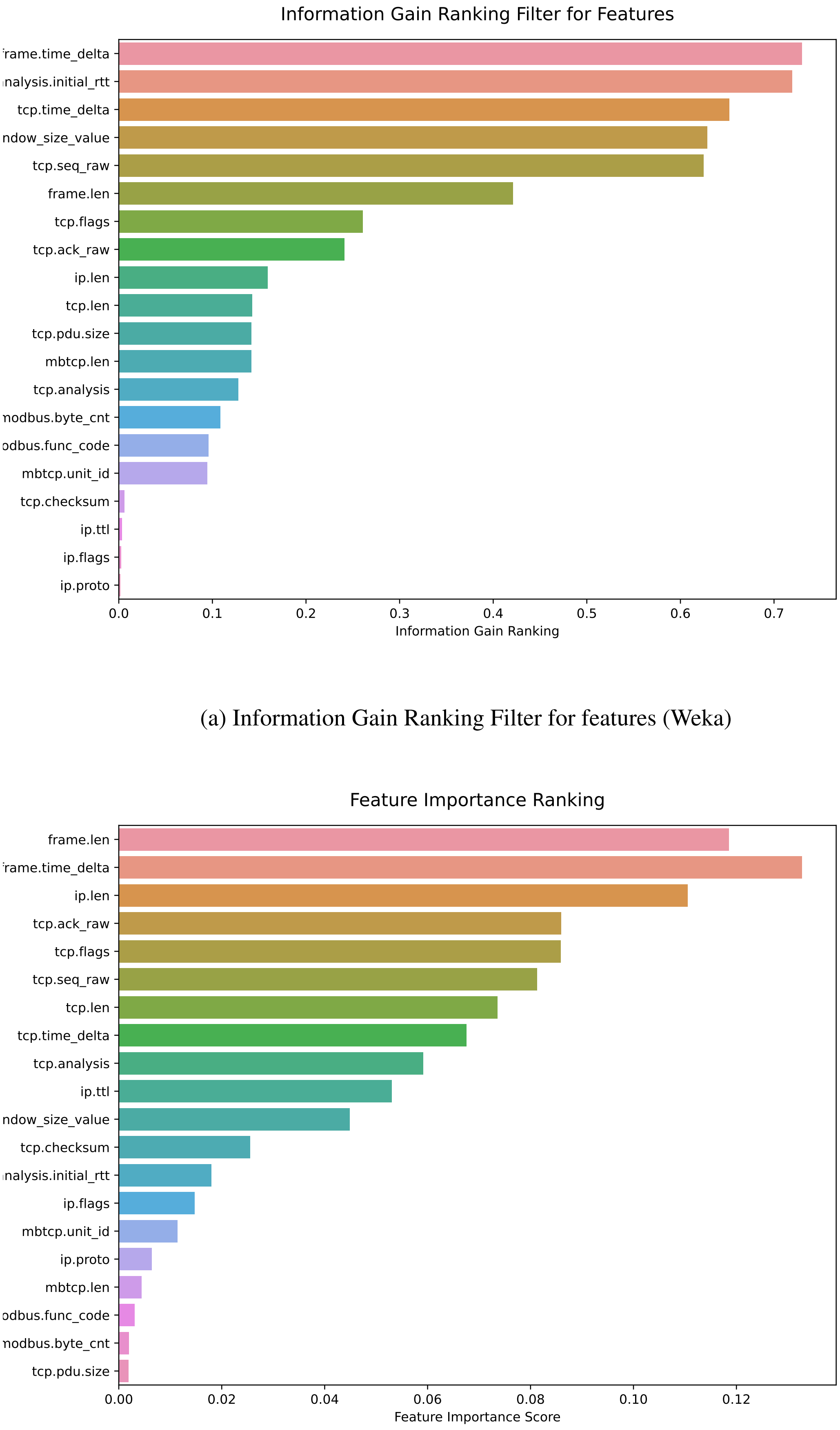
It is of significant note that despite the fact that the dataset used in this Chapter is an expanded version from that of Chapter 4 containing more attacks and variations, the delta time attribute has consistently been identified with a high ranking score. This was also the case in Chapter 4 – indicating that the inter-arrival packet timing, which is a key parameter in determining the periodicity of an industrial network, remains highly relevant in detecting anomalies in the system. On the other hand, the raw SEQ and ACK features have dropped in importance in comparison to results seen in Chapter 4. This could be attributed to the fact that, unlike in the case of the field flooding attack, there are more attacks in the dataset used in this chapter which do not utilise these features in their execution.

5.3.5 Evaluation Criteria

The same evaluation metrics used in chapter 4 would be repeated in these experiments (i.e. precision, recall, F1 score) with the inclusion of 2 additional metrics - the False Positive Rate (FPR) and False Discovery Rate (FDR). These are defined by:

$$FPR = \frac{FP}{FP + TN} \tag{5.1}$$

$$FDR = \frac{FP}{FP + TP} \tag{5.2}$$



(a) Information Gain Ranking Filter for features (Weka)

(b) Feature Importance Ranking Score (scikit-learn)

Figure 5.1: Ranking of most important features using InfoGainAttributeEval and feature_importances_

The false positive rate is the more commonly used metric in the literature. However, the FPR can be misleading when the proportion of malicious instances is extremely low, as is the case in industrial network anomaly detection. This is because the TN is often significantly higher than the TP because majority of the sample data size contains normal data. This will result in a much lower figure for FPR. The FDR on the other hand, considers the TP, which is often less because the anomalous instances are rare when compared to normal instances. For these reasons, in addition to the FPR, we would also compute the FDR. This would aid our understanding of which model performs best in generating the least false alerts.

In the next section, the results of these experiments will be discussed and analysed.

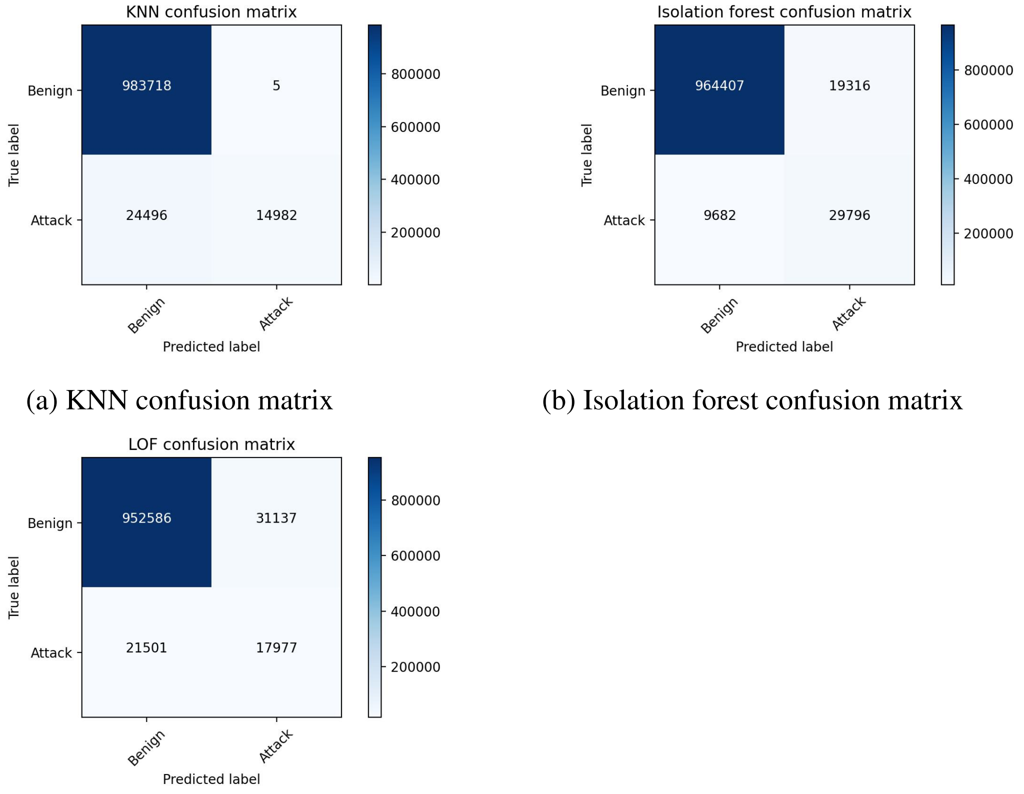
5.4 Results

The results of the experiments showed that the isolation forest algorithm had the highest F1 score of 0.673, with KNN and LOF having F1 scores of 0.55 and 0.455 respectively (Table 5.2). Also, observing the FPR scores, all 3 algorithms obtained impressive numbers with KNN having the best score of **5.08** e^{-6} while isolation forest and LOC scored **0.0196** and **0.0317** respectively. Such low FPR scores (i.e. below 0.04) could be misleading and would suggest that an anomaly detection model is recording relatively low false positives. However, a closer look at the confusion matrices in Figure 5.2 reveals otherwise. For example, the isolation forest and LOC algorithms recorded 19,396 and 31,137 false positives respectively (shown in Figures 5.2b and 5.2c)in the period under consideration (7 hours) while KNN recorded only 5 (Figure 5.2a) in the same period. The FDR metric reflects this performance more accurately with scores of **0.00033**, **0.3933**, and **0.6339** for KNN, isolation forest, and LOC respectively. This means that 39.33% of anomalies detected by the isolation forest algorithm within a 7 hour period were false while for LOC and KNN it was 63.39% and 0.03% respectively. In reality, the FDR metric would be more beneficial than FPR in an operational environment. This is because

Table 5.2: Evaluation metric scores for KNN, isolation forest, and LOC algorithms (for FPR/FDR lower is better)

Algorithm	**Precision**	**Recall**	**F1-Score**	**FPR**	**FDR**
KNN	0.999	0.379	0.55	5.08 e^{-6}	0.00033
Isolation Forest	0.607	0.755	0.673	0.0196	0.3933
LOF	0.366	0.455	0.408	0.0317	0.6339

when processing tens of millions of network data each day, even a modest false discovery rate can overwhelm a security analyst [207] - as can be seen with the high number of false positives recorded by the isolation forest and LOC algorithms within a 7 hour operational period.



(a) KNN confusion matrix

(b) Isolation forest confusion matrix

(c) LOF confusion matrix

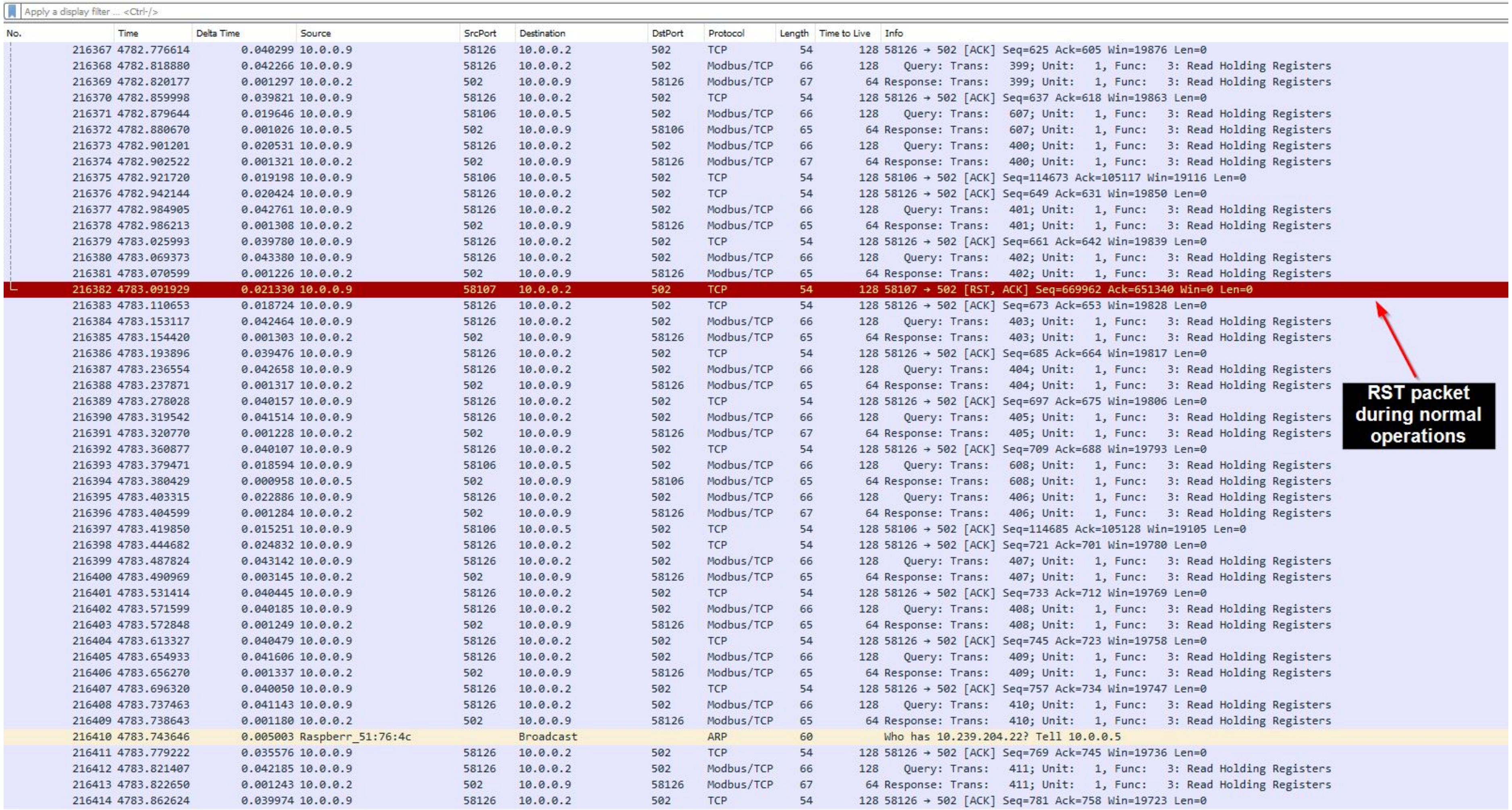
Figure 5.2: Confusion matrices for KNN, isolation forest, and LOF

KNN also had a superior precision score of 0.999 compared to 0.607 and 0.366 scored by isolation forest and LOC respectively. The recall score (also known as the true positive rate or detection rate [208]) is where the KNN lacked a high performance - only scoring 0.379 compared to 0.755 recorded by isolation forest. This suggests that distance-based methods potentially could be leveraged to improve the false positive and false discovery rates. Therefore, investigating new approaches that utilise distance-based methods to detect anomalies with a higher detection rate seems promising.

5.4.1 Summary of Results

Unsupervised anomaly detection is a challenging task - as evidenced by the relatively modest F1 scores obtained by the models. Most unsupervised ML algorithms require assumptions about the data beforehand (i.e. domain, specific application, contamination percentages, and so on). This is certainly one of the fundamental challenges of anomaly detection, brought about mostly by the unsupervised nature of the problem. This assumption is usually in the form of a highly sensitive hyperparameter (e.g. *'contamination'* hyperparameter in isolation forest/LOF) that requires estimating the percentage of anomalies present in the training data. This highly sensitive hyperparameter affects model performance significantly with any slight alteration and would require domain expertise to tune effectively.

Another obvious challenge in anomaly detection methods is trying to reduce the false discovery rates. This is a key element to understand if we are to improve upon its performance. In our case, the models are being asked to analyse each network packet and determine if it belongs to a normal or abnormal traffic class. As an example, we can take a closer examination at instances of a RST packet being transmitted through the network. The most common cause of this is when a SYN packet is sent to a closed port, which could happen during normal operations. From the dataset used, Figures 5.3 and 5.4 show two separate instances of RST packets being transmitted through the network. Figure 5.3 shows a RST packet transmission under normal operations while Figure 5.4 shows several RST packets being transmitted in quick succession (i.e. repetitive pattern) indicative of an anomalous behaviour. With the models assigning binary classifications to each packet, it is harder to differentiate a RST packet occurring under normal circumstances (i.e. Figure 5.3) from those occurring in anomalous circumstances (i.e. Figure 5.4). What may be more beneficial would be to devise a way to capture the behaviour exhibited in Figure 5.4 rather than attempting to classify each packet individually. One way to achieve this could

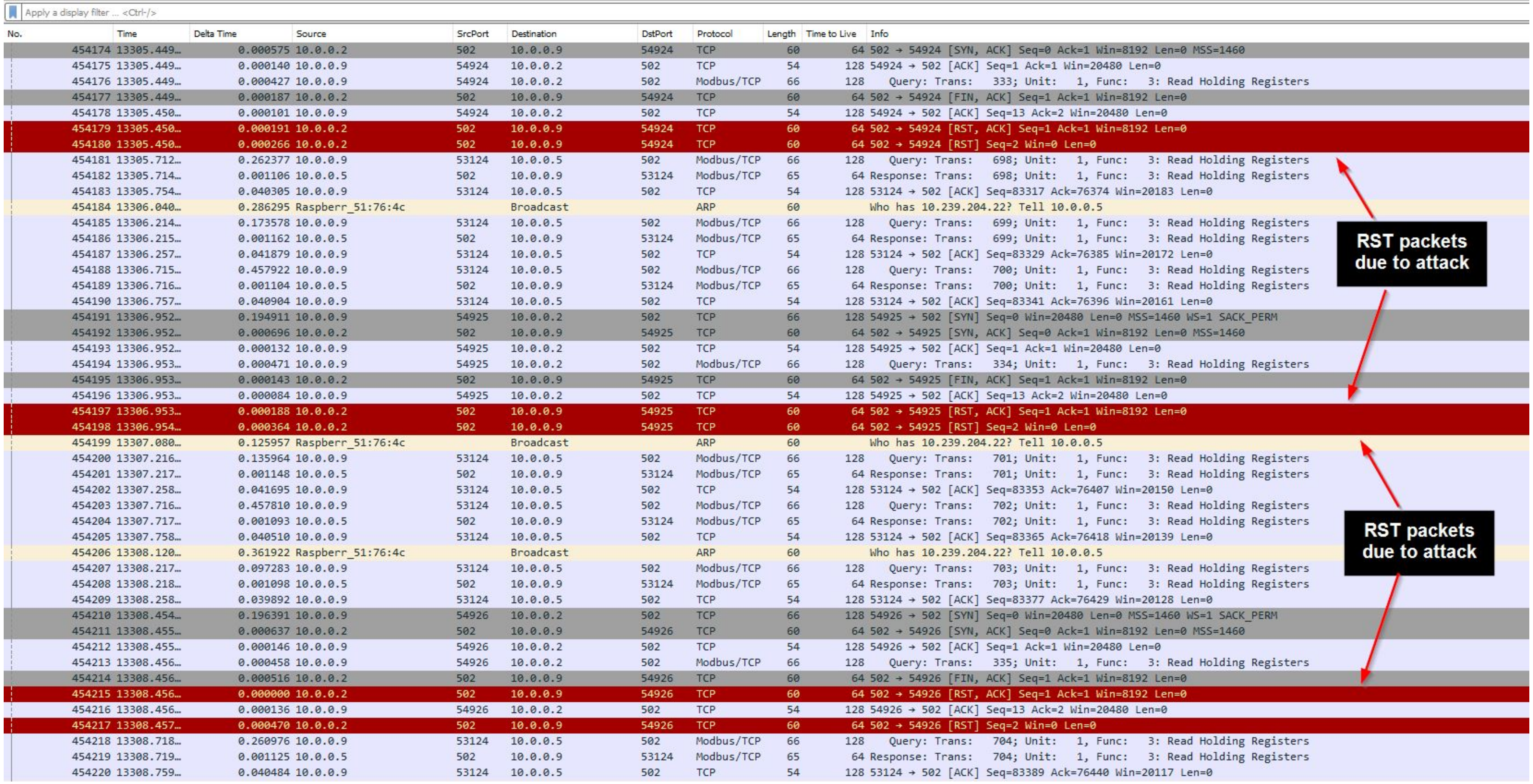


Apply a display filter ... <Ctrl-/>

No.	Time	Delta Time	Source	SrcPort	Destination	DstPort	Protocol	Length	Time to Live	Info
216367	4782.776614	0.040299	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=625 Ack=605 Win=19876 Len=0
216368	4782.818880	0.042266	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 399; Unit: 1, Func: 3: Read Holding Registers
216369	4782.820177	0.001297	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	67	64	Response: Trans: 399; Unit: 1, Func: 3: Read Holding Registers
216370	4782.859998	0.039821	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=637 Ack=618 Win=19863 Len=0
216371	4782.879644	0.019646	10.0.0.9	58106	10.0.0.5	502	Modbus/TCP	66	128	Query: Trans: 607; Unit: 1, Func: 3: Read Holding Registers
216372	4782.880670	0.001026	10.0.0.5	502	10.0.0.9	58106	Modbus/TCP	65	64	Response: Trans: 607; Unit: 1, Func: 3: Read Holding Registers
216373	4782.901201	0.020531	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 400; Unit: 1, Func: 3: Read Holding Registers
216374	4782.902522	0.001321	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	67	64	Response: Trans: 400; Unit: 1, Func: 3: Read Holding Registers
216375	4782.921720	0.019198	10.0.0.9	58106	10.0.0.5	502	TCP	54	128	58106 → 502 [ACK] Seq=114673 Ack=105117 Win=19116 Len=0
216376	4782.942144	0.020424	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=649 Ack=631 Win=19850 Len=0
216377	4782.984905	0.042761	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 401; Unit: 1, Func: 3: Read Holding Registers
216378	4782.986213	0.001308	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 401; Unit: 1, Func: 3: Read Holding Registers
216379	4783.025993	0.039780	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=661 Ack=642 Win=19839 Len=0
216380	4783.069373	0.043380	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 402; Unit: 1, Func: 3: Read Holding Registers
216381	4783.070599	0.001226	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 402; Unit: 1, Func: 3: Read Holding Registers
216382	4783.091929	0.021330	10.0.0.9	58107	10.0.0.2	502	TCP	54	128	58107 → 502 [RST, ACK] Seq=669962 Ack=651340 Win=0 Len=0
216383	4783.110653	0.018724	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=673 Ack=653 Win=19828 Len=0
216384	4783.153117	0.042464	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 403; Unit: 1, Func: 3: Read Holding Registers
216385	4783.154420	0.001303	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 403; Unit: 1, Func: 3: Read Holding Registers
216386	4783.193896	0.039476	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=685 Ack=664 Win=19817 Len=0
216387	4783.236554	0.042658	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 404; Unit: 1, Func: 3: Read Holding Registers
216388	4783.237871	0.001317	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 404; Unit: 1, Func: 3: Read Holding Registers
216389	4783.278028	0.040157	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=697 Ack=675 Win=19806 Len=0
216390	4783.319542	0.041514	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 405; Unit: 1, Func: 3: Read Holding Registers
216391	4783.320770	0.001228	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	67	64	Response: Trans: 405; Unit: 1, Func: 3: Read Holding Registers
216392	4783.360877	0.040107	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=709 Ack=688 Win=19793 Len=0
216393	4783.379471	0.018594	10.0.0.9	58106	10.0.0.5	502	Modbus/TCP	66	128	Query: Trans: 608; Unit: 1, Func: 3: Read Holding Registers
216394	4783.380429	0.000958	10.0.0.5	502	10.0.0.9	58106	Modbus/TCP	65	64	Response: Trans: 608; Unit: 1, Func: 3: Read Holding Registers
216395	4783.403315	0.022886	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 406; Unit: 1, Func: 3: Read Holding Registers
216396	4783.404599	0.001284	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	67	64	Response: Trans: 406; Unit: 1, Func: 3: Read Holding Registers
216397	4783.419850	0.015251	10.0.0.9	58106	10.0.0.5	502	TCP	54	128	58106 → 502 [ACK] Seq=114685 Ack=105128 Win=19105 Len=0
216398	4783.444682	0.024832	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=721 Ack=701 Win=19780 Len=0
216399	4783.487824	0.043142	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 407; Unit: 1, Func: 3: Read Holding Registers
216400	4783.490969	0.003145	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 407; Unit: 1, Func: 3: Read Holding Registers
216401	4783.531414	0.040445	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=733 Ack=712 Win=19769 Len=0
216402	4783.571599	0.040185	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 408; Unit: 1, Func: 3: Read Holding Registers
216403	4783.572848	0.001249	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 408; Unit: 1, Func: 3: Read Holding Registers
216404	4783.613327	0.040479	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=745 Ack=723 Win=19758 Len=0
216405	4783.654933	0.041606	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 409; Unit: 1, Func: 3: Read Holding Registers
216406	4783.656270	0.001337	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 409; Unit: 1, Func: 3: Read Holding Registers
216407	4783.696320	0.040050	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=757 Ack=734 Win=19747 Len=0
216408	4783.737463	0.041143	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 410; Unit: 1, Func: 3: Read Holding Registers
216409	4783.738643	0.001180	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	65	64	Response: Trans: 410; Unit: 1, Func: 3: Read Holding Registers
216410	4783.743646	0.005003	Raspberr_51:76:4c		Broadcast		ARP	60		Who has 10.239.204.22? Tell 10.0.0.5
216411	4783.779222	0.035576	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=769 Ack=745 Win=19736 Len=0
216412	4783.821407	0.042185	10.0.0.9	58126	10.0.0.2	502	Modbus/TCP	66	128	Query: Trans: 411; Unit: 1, Func: 3: Read Holding Registers
216413	4783.822650	0.001243	10.0.0.2	502	10.0.0.9	58126	Modbus/TCP	67	64	Response: Trans: 411; Unit: 1, Func: 3: Read Holding Registers
216414	4783.862624	0.039974	10.0.0.9	58126	10.0.0.2	502	TCP	54	128	58126 → 502 [ACK] Seq=781 Ack=758 Win=19723 Len=0

RST packet during normal operations

Figure 5.3: RST packet occurrence during normal operations



```
Apply a display filter ... <Ctrl-/>
No.     Time         Delta Time Source            SrcPort Destination DstPort Protocol   Length Time to Live Info
 454174 13305.449…   0.000575 10.0.0.2            502     10.0.0.9    54924   TCP        60      64 502 → 54924 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
 454175 13305.449…   0.000140 10.0.0.9            54924   10.0.0.2    502     TCP        54     128 54924 → 502 [ACK] Seq=1 Ack=1 Win=20480 Len=0
 454176 13305.449…   0.000427 10.0.0.9            54924   10.0.0.2    502     Modbus/TCP 66     128    Query: Trans:   333; Unit:   1, Func:   3: Read Holding Registers
 454177 13305.449…   0.000187 10.0.0.2            502     10.0.0.9    54924   TCP        60      64 502 → 54924 [FIN, ACK] Seq=1 Ack=1 Win=8192 Len=0
 454178 13305.450…   0.000101 10.0.0.9            54924   10.0.0.2    502     TCP        54     128 54924 → 502 [ACK] Seq=13 Ack=2 Win=20480 Len=0
 454179 13305.450…   0.000191 10.0.0.2            502     10.0.0.9    54924   TCP        60      64 502 → 54924 [RST, ACK] Seq=1 Ack=1 Win=8192 Len=0
 454180 13305.450…   0.000266 10.0.0.2            502     10.0.0.9    54924   TCP        60      64 502 → 54924 [RST] Seq=2 Win=0 Len=0
 454181 13305.712…   0.262377 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   698; Unit:   1, Func:   3: Read Holding Registers
 454182 13305.714…   0.001106 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   698; Unit:   1, Func:   3: Read Holding Registers
 454183 13305.754…   0.040305 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83317 Ack=76374 Win=20183 Len=0
 454184 13306.040…   0.286295 Raspberr_51:76:4c           Broadcast           ARP        60        Who has 10.239.204.22? Tell 10.0.0.5
 454185 13306.214…   0.173578 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   699; Unit:   1, Func:   3: Read Holding Registers
 454186 13306.215…   0.001162 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   699; Unit:   1, Func:   3: Read Holding Registers
 454187 13306.257…   0.041879 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83329 Ack=76385 Win=20172 Len=0
 454188 13306.715…   0.457922 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   700; Unit:   1, Func:   3: Read Holding Registers
 454189 13306.716…   0.001104 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   700; Unit:   1, Func:   3: Read Holding Registers
 454190 13306.757…   0.040904 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83341 Ack=76396 Win=20161 Len=0
 454191 13306.952…   0.194911 10.0.0.9            54925   10.0.0.2    502     TCP        66     128 54925 → 502 [SYN] Seq=0 Win=20480 Len=0 MSS=1460 WS=1 SACK_PERM
 454192 13306.952…   0.000696 10.0.0.2            502     10.0.0.9    54925   TCP        60      64 502 → 54925 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
 454193 13306.952…   0.000132 10.0.0.9            54925   10.0.0.2    502     TCP        54     128 54925 → 502 [ACK] Seq=1 Ack=1 Win=20480 Len=0
 454194 13306.953…   0.000471 10.0.0.9            54925   10.0.0.2    502     Modbus/TCP 66     128    Query: Trans:   334; Unit:   1, Func:   3: Read Holding Registers
 454195 13306.953…   0.000143 10.0.0.2            502     10.0.0.9    54925   TCP        60      64 502 → 54925 [FIN, ACK] Seq=1 Ack=1 Win=8192 Len=0
 454196 13306.953…   0.000084 10.0.0.9            54925   10.0.0.2    502     TCP        54     128 54925 → 502 [ACK] Seq=13 Ack=2 Win=20480 Len=0
 454197 13306.953…   0.000188 10.0.0.2            502     10.0.0.9    54925   TCP        60      64 502 → 54925 [RST, ACK] Seq=1 Ack=1 Win=8192 Len=0
 454198 13306.954…   0.000364 10.0.0.2            502     10.0.0.9    54925   TCP        60      64 502 → 54925 [RST] Seq=2 Win=0 Len=0
 454199 13307.080…   0.125957 Raspberr_51:76:4c           Broadcast           ARP        60        Who has 10.239.204.22? Tell 10.0.0.5
 454200 13307.216…   0.135964 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   701; Unit:   1, Func:   3: Read Holding Registers
 454201 13307.217…   0.001148 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   701; Unit:   1, Func:   3: Read Holding Registers
 454202 13307.258…   0.041695 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83353 Ack=76407 Win=20150 Len=0
 454203 13307.716…   0.457810 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   702; Unit:   1, Func:   3: Read Holding Registers
 454204 13307.717…   0.001093 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   702; Unit:   1, Func:   3: Read Holding Registers
 454205 13307.758…   0.040510 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83365 Ack=76418 Win=20139 Len=0
 454206 13308.120…   0.361922 Raspberr_51:76:4c           Broadcast           ARP        60        Who has 10.239.204.22? Tell 10.0.0.5
 454207 13308.217…   0.097283 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   703; Unit:   1, Func:   3: Read Holding Registers
 454208 13308.218…   0.001098 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   703; Unit:   1, Func:   3: Read Holding Registers
 454209 13308.258…   0.039892 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83377 Ack=76429 Win=20128 Len=0
 454210 13308.454…   0.196391 10.0.0.9            54926   10.0.0.2    502     TCP        66     128 54926 → 502 [SYN] Seq=0 Win=20480 Len=0 MSS=1460 WS=1 SACK_PERM
 454211 13308.455…   0.000637 10.0.0.2            502     10.0.0.9    54926   TCP        60      64 502 → 54926 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
 454212 13308.455…   0.000146 10.0.0.9            54926   10.0.0.2    502     TCP        54     128 54926 → 502 [ACK] Seq=1 Ack=1 Win=20480 Len=0
 454213 13308.456…   0.000458 10.0.0.9            54926   10.0.0.2    502     Modbus/TCP 66     128    Query: Trans:   335; Unit:   1, Func:   3: Read Holding Registers
 454214 13308.456…   0.000516 10.0.0.2            502     10.0.0.9    54926   TCP        60      64 502 → 54926 [FIN, ACK] Seq=1 Ack=1 Win=8192 Len=0
 454215 13308.456…   0.000000 10.0.0.2            502     10.0.0.9    54926   TCP        60      64 502 → 54926 [RST, ACK] Seq=1 Ack=1 Win=8192 Len=0
 454216 13308.456…   0.000136 10.0.0.9            54926   10.0.0.2    502     TCP        54     128 54926 → 502 [ACK] Seq=13 Ack=2 Win=20480 Len=0
 454217 13308.457…   0.000470 10.0.0.2            502     10.0.0.9    54926   TCP        60      64 502 → 54926 [RST] Seq=2 Win=0 Len=0
 454218 13308.718…   0.260976 10.0.0.9            53124   10.0.0.5    502     Modbus/TCP 66     128    Query: Trans:   704; Unit:   1, Func:   3: Read Holding Registers
 454219 13308.719…   0.001125 10.0.0.5            502     10.0.0.9    53124   Modbus/TCP 65      64 Response: Trans:   704; Unit:   1, Func:   3: Read Holding Registers
 454220 13308.759…   0.040484 10.0.0.9            53124   10.0.0.5    502     TCP        54     128 53124 → 502 [ACK] Seq=83389 Ack=76440 Win=20117 Len=0
```

RST packets due to attack

RST packets due to attack

Figure 5.4: Anomalous occurrences of RST packets

be using time windows where the sequence of packets can be better analysed. This would open the path to analysing the network behaviour rather than the individual packet.

5.5 Conclusions

This chapter assessed the performance of three anomaly detection methods to detect field flooding, MITM, and SYN flooding attacks in an industrial network using unsupervised machine learning. The aim was partly to evaluate model performance and identifying which of the models generates the least false alerts. This was important because one of the major challenges with unsupervised ML anomaly detection is the generation of high false positives. The key to industry adoption of these methods lies in the development of an unsupervised method that consistently generates low false alerts. To help with this investigation, 3 popular unsupervised algorithms - KNN, isolation forest, and LOF - were trained using data collected from the gas wellhead monitoring station testbed described in chapter 4. The algorithm that had the highest F1 score was isolation forest, obtaining a score of 0.673 while KNN and LOC scored 0.55 and 0.408 respectively.

However, analysing the false discovery rates showed that in a 7 hour period (duration of the dataset), 39.33% and 63.39% of anomalies detected by the isolation forest and LOC algorithms respectively were false positives while the KNN FDR score of 0.03% was a far superior performance comparatively. KNN also had the best precision score of 0.999 compared to 0.755 and 0.366 obtained by isolation forest and LOC respectively. The FDR metric in anomaly detection is important in operational environments because even a modest false discovery rate can overwhelm security analysts. The aim therefore is to improve current methods towards achieving anomaly detection accuracy of zero FDR. This chapter also showed that the FDR metric is more useful in evaluating the performance of anomaly detection models in an industrial environment than the FPR which is more predominantly used in the literature.

Considering that the KNN method recorded the best FDR and precision scores, it would be useful to investigate its approach further to improve its detection rate and develop a more effective unsupervised anomaly detection method. The KNN method utilises a distance function like the euclidean distance function (L2-norm) to determine the distance between the data and their k^{th} nearest neighbour to quantify the degree of the anomaly.

Significantly, similar to Chapter 4, this chapter showed that applying feature importance ranking filters on the expanded dataset revealed that the delta time attribute has consistently ranked high in relevance. This is potentially an indication that the packet inter-arrival time – which is a key parameter used to determine the periodicity of an industrial network – is critical in determining deviations from normal behaviour in an industrial network. This validates the next phase of our investigations (explored in Chapter 6) which is focused on anomaly detection using the high periodicity characteristic of industrial networks.

Furthermore, the anomaly detection methods investigated in this chapter were designed to analyse each packet in the network transmission and classify it as either a normal or anomalous packet. This could be one of the reason for the high FDR scores recorded by the models. Therefore, one potential way of improving performance could be to utilise time windows. This would enable analysing network behaviour patterns rather than the more difficult task of classifying each individual packet - resulting in high false positives. In the next chapter, this direction will be explored further.

Chapter 6

STADe: An Unsupervised Time-Windows Method of Detecting Anomalies in Oil and Gas Industrial Cyber-Physical Systems (ICPS) Networks

Parts of this chapter are included in the paper "STADe: An Unsupervised Time-Windows Method of Detecting Anomalies in Oil and Gas Industrial Cyber-Physical Systems (ICPS) Networks" currently under review

6.1 Introduction

The results in the previous chapter indicate that unsupervised methods, though capable of detecting sufficient anomalies, have a limitation of high False Discovery Rate (FDR). However, distance-based methods show promise with very low FDR. Also, recall that in Chapter 2, we highlighted that industrial networks exhibit a strong periodic behaviour due to machine-to-machine interactions. This behaviour was affirmed from the feature ranking carried out in experiments in Chapters 4 and 5 as the delta time feature was consistently ranked high. This was a further indication that packet timing as a feature had significant impact on the target variable and was highly correlated with it. Furthermore, another potential strategy identified towards improving the FDR was the incorporation of time windows. Employing time windows offers the advantage of scrutinizing network traffic behavior, alleviating the more intricate challenge of individually classifying each packet. This approach necessitates an investigation of methods for determining and measuring the periodicity of industrial traffic, coupled with the application of distance-based metrics.

Therefore, the objective of this chapter is to explore and formulate an unsupervised anomaly detection approach, capitalising on the high periodicity inherent in industrial network communications. Additionally, we seek to enhance the detection performance by leveraging distance-based methods to minimise the FDR and False Positive Rate (FPR).

As mentioned in Section 2.6.3, the periodic nature of industrial network communication is what makes anomaly detection promising as a means of securing critical infrastructure. Investigating how to utilise this periodicity in anomaly detection, provides the answer to our fifth research question:

RQ5 *How can the inherent periodicity within industrial networks be effectively leveraged to enhance anomaly detection?*

Through answering this question, this chapter will provide the seventh contribution:

C7 *This research contributes a novel methodology of unsupervised Time-Series Method of Detecting Anomalies in Industrial Cyber-Physical Systems (ICPS) Networks.*

The experiments carried out in this chapter are motivated by the observation that industrial control systems and PLC-based systems have a high degree of periodicity in behaviour when used in a real-world context compared to other Information Technology (IT)-based traffic [17].

In this chapter, we present STADe - a Sliding Time-window Anomaly Detection method that uses a sliding window to characterise the periodicity of any given industrial network using the packet timings to create a mini-model of the system which represents the normal operation pattern. This periodic pattern allows the use of anomaly detection to determine where the periodicity is broken (e.g. injection of cyber-attacks). This work is focused on a computationally efficient mechanism to identify this break in periodicity, which flags anomalies and detects potential cyber-attacks. Previous studies on industrial network anomaly detection have mostly focused on inspecting individual packets through deep packet inspection (DPI), machine learning models, or other custom modules which can be error-prone – as evidenced by the high false positive rates experienced by these methods [209, 210]. The advantage of using a time window, comprising multiple packets, is that it can adequately capture network behaviour over a specified period of time, and thus, is more effective at labeling a particular series of packets as either normal or anomalous with a much lower false positive rate when compared with trying to label a specific individual packet.

The remainder of this chapter is structured as follows: Section 6.2 discusses timing-based anomaly detection models in this research area. Section 6.3 describes the approach and STADe methodology. In Section 6.4 description of the experiments carried out is given, while Section 6.5 presents the results and discusses them in more detail. How STADe could be utilised to mitigate earlier identified vulnerabilities in OG operations is highlighted in Section 6.6 and the conclusion is in Section 6.7.

6.2 Timing-Based Anomaly Detection

Despite the importance of critical infrastructure, and the increasing threats to it from cyber attacks, only a few studies have investigated anomaly detection methods focused on the high periodicity of industrial networks. Many current network anomaly detection systems are based on supervised machine learning methods which are often expensive and difficult to obtain training data [108, 211], while unsupervised machine learning anomaly detection methods are are not widely used in practice because of high false positive rates that tend to overwhelm security analysts with false alerts. To address these problems, researchers have recently turned to deep learning techniques, which have also increased computational overheads. Industrial control networks require lightweight solutions and, as such, for these reasons, in this chapter, we have not considered machine learning and deep learning-based anomaly detection methods.

In other recent anomaly detection approaches for industrial networks, such as [212], the authors used a timing-based anomaly detection system that uses the statistical attributes of the communication patterns. Their proposed intrusion detection system identifies unique sets of request-response events from request types and requested addresses. Jiang et al. [213] also proposed a method to detect network traffic anomalies by using a sliding window that uses Decomposable Principal Component Analysis (DPCA) to handle the traffic of all original destination flows in a network. The method comprises utilising compressed features of the network traffic including byte size to classify anomalous scenarios and was evaluated using traffic data from the Abilene network. By incorporating addresses into their learning modules, the models in [212] and [213] would struggle to detect stealthier attacks like MITM which spoofs legitimate IP addresses morphed into the data stream.

In [214], Tekeoglu et al. used network traffic features to capture the system-specific anomalies. They did this by extracting a number of packet-based features from pcap files using 3-second time windows which included average bytes in the window, average seconds between each consecutive packet in a window, and unique destination IP addresses in the window amongst others. This method required constant iterations between network traffic features to determine the most appropriate metric and, as a result, increases its computational overhead.

The authors in [154] and [215] also tried to utilise the traffic periodicity in industrial networks by using message repetition and timing information to automatically learn traffic models that capture periodic patterns. The authors in [154] proposed a period analyser composed of three modules: (i) Multiplexer, (ii) Tokeniser, and (iii) Learner which worked in a sequence comprising preprocessing the network traffic and separating it into different flows, transforming each packet into a protocol-independent format called a token, and finally processing each token to identify and characterise periodic activities called cycles. Their method involves filtering and grouping packets based on server address, IP protocol, server port, and client address. One key limitation of their study is that the tokeniser and multiplexer modules need to be adapted in order to accommodate new industrial protocols as it was designed for Modbus and MMS protocols. The authors in [215] also relied on the stable and persistent control flow communications in industrial networks to develop a fingerprinting methodology to capture normal behaviour characteristics. They extracted multiple features such as packet arrival order, packet size, direction, and inter-arrival time to classify network behaviour. This requirement of using multiple features from network traffic increases the computational head on the system.

In general, industrial network anomaly detection requires a lightweight solution. The more features the IDS model utilises in its detection engine, the higher the computational overhead on the system. Although some of these approaches utilise time windows which could potentially improve efficiency because computation is done on a subset of data at a time, having multiple features still increases complexity to some extent. All of the methods mentioned in this section utilise multiple features from network traffic in order to train a learner module that would subsequently classify the packet as normal or otherwise. This is all summarised in Table 6.1.

By using packet inter-arrival times as the only feature, STADe offers a method that has less computational overhead that can potentialy operate in an industrial network environment. Another advantage of the STADe approach is that because it only uses delta times as a single feature, it can potentially be deployed alongside other encryption-based security solutions to improve protection. This is usually a pitfall for most anomaly detection engines because after encryption, depending on the encryption methods deployed, the network data (e.g. source/destination addresses, source/destination ports, and data payloads) required for analyses would no longer be useful to the detection models. This limitation, which restricts the deployment of multiple approaches to secure an system, does not apply to STADe.

Table 6.1: Summary of timing-based approaches

Reference	**Data source from real hardware**	**Utilises Time Windows**	**Single feature-based**	**Zero False Positives**
[212]	●			
[213]	●	●		
[214]	●	●		
[154]	●			
[215]	●			
STADe	●	●	●	●

In the next section, we describe the STADe approach and methodology in detail, showing how it could be used to characterise and subsequently detect industrial network traffic anomalies.

6.3 STADe: Approach and Methodology

6.3.1 Measuring Periodicity

To identify breaks in periodicity in a given industrial network, we first have to be able to measure network traffic periodicity. This involves the use of data in the form of ordered observations with respect to time. To increase ease of computation, some features of the data may be neglected, and thus only analysing the time between events [216]. If only the time of the event (i.e. data/packet transmission) and no further details are stored, the event sequence is called a point sequence [216]. Several studies, such as [217], have done this by capturing number of packets per second as a feature, while others (e.g. [218]) used arrival time of packets as its core feature. This not only underscores the importance that packet timing, observed as point sequences, has in defining the periodic nature of industrial network traffic but also shows its usefulness in getting information from network traffic [219].

Hubballi et al. [220] posited that periodic communications exhibit very low variance and standard deviation considering their inter-time differences and determined this low variance by taking the standard deviation of packet inter-arrival times between network packets. We adapted this approach in our study by calculating the standard deviation of packet inter-arrival times within a time window of packets.

Thus, the standard deviation of a window (SD_w) gives the variance or level of dispersion within that distribution. This is for any given window with each element in the distribution (x_i), sample mean ($\bar{x}$), and window size (n).

$$SD_w = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \tag{6.1}$$

6.3.2 Detecting Deviation from Periodicity

The advantage of being able to easily measure the periodicity of an industrial network is to gain the ability to detect a break or deviation from the periodic pattern. To do this, the data feature collected (i.e. packet inter-arrival times) can be segmented into time windows of the same size. These time windows can then be compared to one another to determine their similarity (normal traffic) or dissimilarity (anomaly). We later describe a metric - the diff score - which will compute if the data is anomalous or not. One way to do this is to select a representative time window (baseline/sliding time window), which can then be compared to all other windows using some distance function. The authors in [221] did this by computing the average trend in a segment that minimises the sum of distances to the other segments - in other words, computing the euclidean distances. We have adopted this approach, similar to the KNN approach in Chapter 5, to compute the euclidean distance between the average trend of a time window distribution and that of the sliding window under consideration.

This is achieved by computing the L2-norm as it is the estimate of location that minimises the euclidean distance between two time windows and is represented as the diff centre (DC_w) for a given window (i), window size (n), and sliding window (j).

$$DC_w(i,j) = \sqrt{\Sigma_{i=1}^{n}(i-j)^2} \tag{6.2}$$

6.3.3 Visualising Periodic Behaviour

Recording the datastream from an industrial network as point events allows us to represent the packet inter-arrival times visually as points in 3-dimensional space, similar to the approaches in [222] and [219] where the authors try to classify network traffic visually. In [219], bigrams of packets are visualised where the coordinates are defined by the first packet's size (X), the inter-arrival time between the two packets (Y), and the second packet's size (Z), while the coordinates in [222] represent sequence number, frame length, and packet number to classify telecommunications traffic.

The STADe visualisation approach differs from previous studies because each coordinate represents information from only the inter-arrival times between three consecutive packets. The advantage of using 3-dimensional spaces over 2D is that just one coordinate can be used to display information of more packets (i.e. 3 packets) and would require less space to represent a network's basic pattern than 2-dimensional space would.

The packet flow regularity in industrial networks is the characteristic that enables a sliding window with a single feature of packet timings to represent the basic normal pattern of operations in a given industrial system. This allows us to compute any significant deviations from the basic pattern represented in the sliding window to detect anomalous scenarios within the network.

6.3.4 Methodology

Based on the described approach, the framework of the STADe methodology (shown in Figure 6.1) is described as follows:

1. Extract packet inter-arrival times δ from a Cyber-Physical System data stream as a vector.

2. Divide the extracted δ into windows (W) of same segment size, n, such that δ =

$(W_1, W_2, ..., W_m)$ with elements $t_i \in W$. A fixed segmentation $\delta(n)$ of size n is a division of δ into m windows where each of the windows consists of n consecutive elements from δ.

3. Select baseline/sliding window containing normal traffic.
4. Calculate the standard deviation and diff centres for each window.
5. Compare the sliding window with other windows by computing the diff score S. The diff score is a combination of the standard deviation and diff centres using a weight x ranging from 0 - 1. This essentially allows the flexibility of assigning more weight (importance) to either the standard deviation or diff centre (i.e. weight tends towards 0 or 1 respectively) or assigning equal importance to both (i.e. weight = 0.5).
6. The diff scores of all window segment comparisons $S_\delta = (S_{w_1}, S_{w_2}, ..., S_{w_n})$ is generated and stored as an array.

In the next section, a detailed description of the experiments carried out to validate the STADe methodology is given with emphasis on the hyper-parameters, dataset collection, and visualisation methods.

6.4 Experiments

6.4.1 Dataset collection

For these experiments, the same dataset described in Chapter 5 containing field flooding, SYN flooding, and MitM attacks will be used to evaluate the STADe methodology. However, to effectively measure the performance on each attack, the dataset is split into three – isolating each attack category in a separate dataset. An additional dataset containing only normal traffic to enable determination of the threshold is also collected. This makes it a total of four (4) datasets collected from the gas wellhead monitoring testbed described in Chapter 4 to enable evaluation of the performance of STADe on detection of each attack. Furthermore, all datasets were designed to have a similar attack time pattern where the

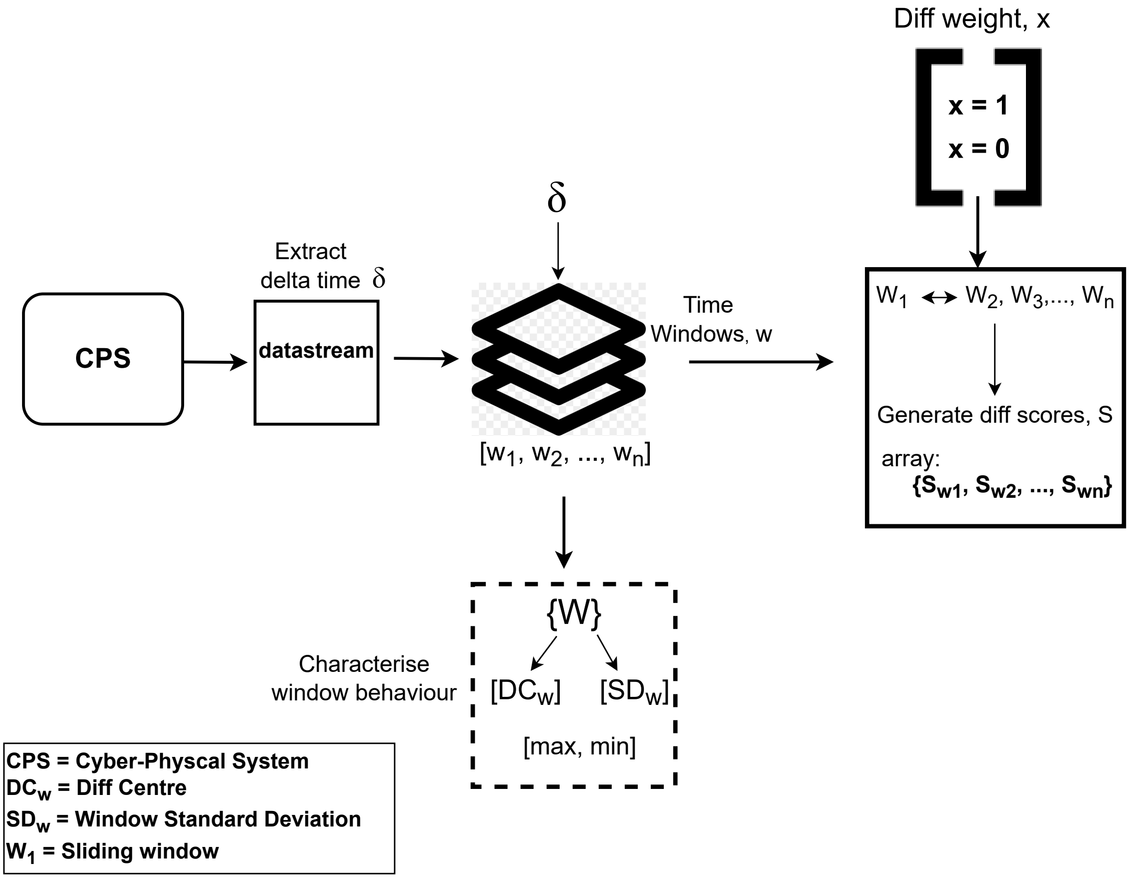


Figure 6.1: STADe methodology

first 80% represents normal traffic, an attack is executed in the next 10%, and the final 10% of the network traffic is normal as illustrated in Figure 6.2. This helps us to have a general idea of where our model should be detecting attacks and helps with evaluating its performance.

It is important to note that a successful MITM attack could serve as an initial phase which makes a number of further attacks possible in a second phase. A list of these possible second phase attacks are highlighted in Table 6.2. The summary of all the datasets is also shown in Table 6.3

6.4.2 Visualising entire datasets

The initial phase of the experiments entailed creating a 3D plot of the network packets under normal operations to visually investigate if there is a set pattern that could represent network behaviour. To do this, normal traffic data was collected from the testbed over a 22-hour period. The resulting dataset contained 3,409,005 packets. A second dataset was

Table 6.2: Further potential attacks after successful MITM attack

1st phase Attack	**2nd Phase Attack**	**Impact**
MITM	Data interception	Attacker can eavesdrop on communication between HMI and PLC, capturing sensitive commands and process state
	Data tampering	Attacker can modify data transmitted to PLC, altering commands
	Session hijacking	Attacker can hijack an established session between two parties and taking control by impersonating one of the legitimate parties
	Credential theft	Admin login credentials could be intercepted, enabling lateral movement within a network
	Replay attacks	Attacker can capture data packets and replay them later, potentially sending same commands but in the wrong context
	Malware delivery	Attacker can utilise MITM position to deliver malicious software into victim's network or devices
	DNS spoofing	Attacker can manipulate DNS responses, or intercept DNS queries within the network,potentially redirecting requests to malicious end points
	Phishing	After intercepting communication, the attacker can launch further phishing campaigns with targeted information. An example could be to target remote engineers accessing process systems

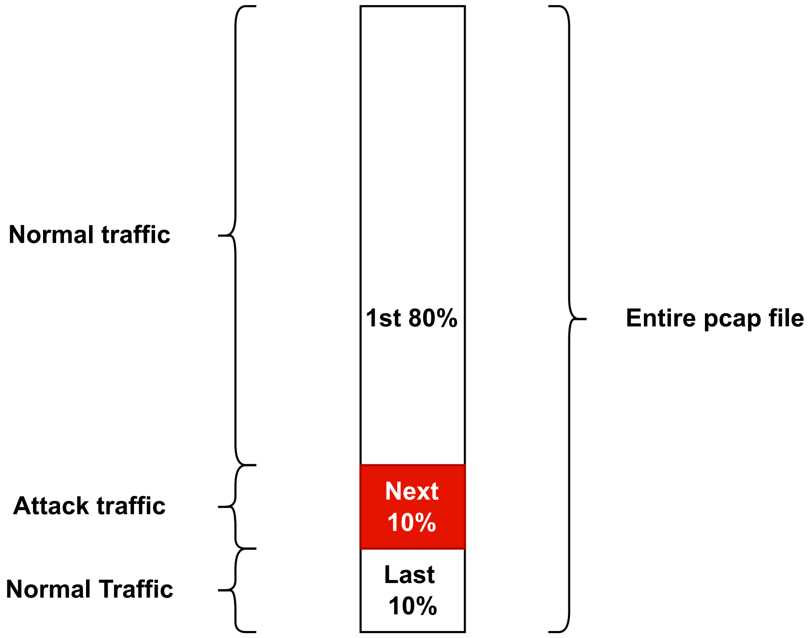


Figure 6.2: Dataset creation

collected from the testbed comprising just 3.8 hours of normal traffic + field flooding attack [223]. This second dataset contained 472,887 packets. Both plots are shown in Figure 6.3.

From Figure 6.3, although a pattern can be observed, it seems distorted with a lot of noise. This is the primary limitation of plotting an entire dataset on a single plot because over 90% of the data points lie within the dense circled area. As a result, the scale of the plot is larger than it needs to be to accommodate anomalous coordinates. It becomes evident that to see deeper underlying patterns that could potentially lie within the dense circled area, a smaller-scaled plot representing a window (or subset) of network traffic would be more beneficial. This confirms the advantage of using the STADe methodology described in Section 6.3.4 to create a smaller sliding window, compare it with other windows and generate the diff scores between them.

Despite this limitation, the plot is still useful because when visually inspecting Figures 6.3a and 6.3b, there are some obvious distortions in the network pattern seen in 6.3b that were not evident in 6.3a. These differences could be as a result of the field flooding attack but can not be confirmed visually. In this case, having a numerical score as a basis for comparison and evaluation would be more beneficial. This numerical score is represented

Table 6.3: Summary of datasets collected from testbed

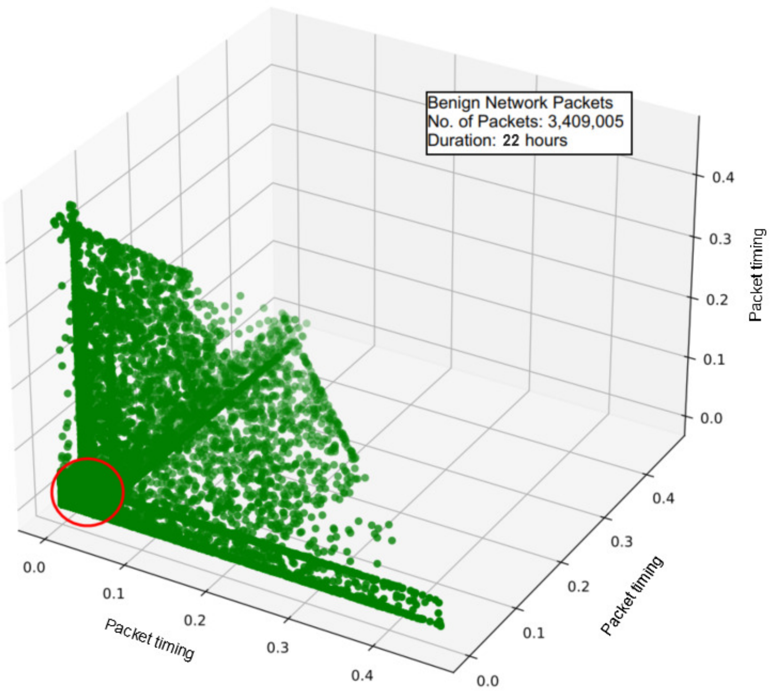
Dataset	**Attack**	**Attack Type**	**Target**	**Attack Duration**	**Dataset Duration**	**Total No. Pkts**
Dataset 1 (normal traffic)	N/A	N/A	N/A	N/A	22.4 hours	3,409,005
Dataset 2	Field Flooding	DoS	Availability	1 hour	3.8 hours	472,887
Dataset 3	SYN Flooding	DDoS	Availability	13 secs	1.4 hours	230,409
Dataset 4	MITM	See Table 6.2	Integrity	5.5 mins	2.1 hours	319,906

by the diff score described in the STADe methodology.

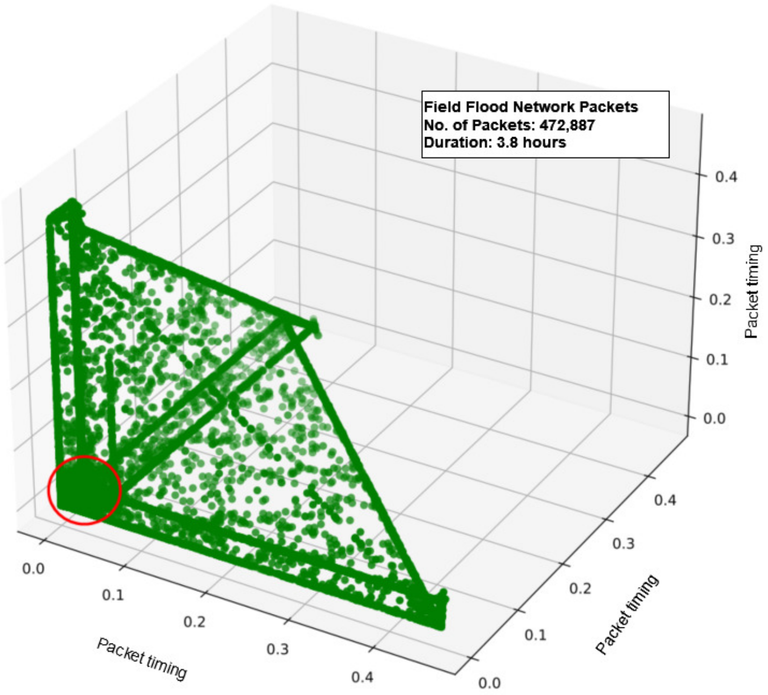
6.4.3 Hyper-parameters:

The next phase of the experiments to be carried out is to generate diff scores for all 4 datasets with the following hyper-parameters:

- Window size, n: Industrial networks have communication cycles (i.e. query-response-acknowledgement cycle). Each device is queried sequentially at least once until all devices have responded. Then the communication would loop and start all over. To determine an adequate window size, a significant amount of loops (repetitions) containing all normal operating conditions should be captured within it. In this study, we investigated window sizes containing 30, 60, 90, and 120 seconds of network communication. An optimal window size of 60 seconds (1 minute of network communication) was determined empirically. This contained approximately 2500 packets and more significantly, captured all communicating devices and their respective commands. While determining the window size, it was observed that below 60 seconds, the window did not capture enough information to characterize network behaviour, hence this resulted in some normal operations being flagged as anomalous (i.e. more false positives). Similarly, above 60 seconds, the window size contained more information than was necessary, which resulted in some anomalous



(a) 22-hour normal operations Network traffic - Packet inter-arrival times



(b) 3.8-hour normal+field flooding attack Network traffic - Packet inter-arrival times

Figure 6.3: 3D plots for Network traffic packet inter-arrival times

behaviours being classified as normal (i.e. more false negatives).

It is important to note that this approach may vary in larger and more complex industrial environments.

- Diff Score weight: The diff score weight, ranging from 0 - 1, is the importance given to measurements of diff centres and areas of coverage. A diff weight of 0.5 was chosen to give equal importance to either parameter.

- Threshold: The next important parameter is the definition of a threshold. To define a suitable threshold, diff scores need to be generated using normal traffic data. The aim is to capture all possible usual network scenarios which include random packet retransmissions, legitimate connection resets, etc. The maximum diff score generated from normal traffic can be used as a valid threshold. The amount of normal traffic required would vary depending on the size of the industrial network. In our case, we monitored normal network traffic (with no attacks) for 22 hours and collected the data. Diff scores were generated for the whole dataset containing normal traffic and the maximum diff score was selected as follows:

$$maxdiffscore_{normaloperations} = 0.00216024 = threshold$$

After defining the hyper-parameters, the first window (i.e. first 2,500 packets), which had already been pre-determined to be a normal traffic window, was selected as the baseline window (i.e. sliding window). Recall that all subsequent windows will be compared with the baseline window, and a diff score generated to measure the differences. Any diff score (difference) above our determined threshold would be flagged as an anomaly. For this study, any window within the first 80% (or last 10%) of the dataset could have been chosen as our baseline window. This decision does not affect the results so the user can be flexible in their choice of a baseline/sliding window as long as it is a normal traffic window.

6.4.4 Evaluation Metrics

To evaluate the performance of STADe on our labeled datasets, the same metrics used in Chapter 5 were chosen (i.e. precision, recall, F1 score, FDR, and FPR). The results of the generated diff scores for our attack datasets will be discussed in the next section.

6.5 Results

Diff scores were generated for all three (3) attack datasets using the hyper-parameters determined in Section 6.4. We discuss each attack separately in the following subsections.

6.5.1 Selection and visualisation of baseline window

The first step was to visualise our baseline window. This becomes our sliding window and a sort of label for the dataset that would be compared with every other window by generating a diff score. It can also serve as a basis for a visual comparison with any other window identified as having a diff score higher than the set threshold. As an example, we compared this baseline window to another normal traffic window and generated a diff score between them to establish a correlation. Our baseline window indexes were 0 - 2,499 (i.e. first 2,500 packets) while the random window comparison indexes were x - (x + 2,499), where x = 6,000 (i.e. the 6,000th packet). This procedure was repeated for all three datasets containing attacks and the results obtained are discussed in the next subsections.

6.5.2 Diff Score generation for Dataset 2 (Field Flooding attack):

The baseline window and random window plots are shown in Figures 6.4a and 6.4b respectively while a diff score of **0.000698620** was generated from their comparison. Visually inspecting both plots (i.e. Figures 6.4a and 6.4b), a similar pattern can be observed, although, it is not conclusive. However, their similarity, when evaluated mathematically using the diff score, was confirmed as the result was significantly below the threshold of **0.00216024**. This method confirms that the random window (i.e. packet index 6,000 - 7,499), when compared with the baseline window contains normal traffic.

Next, the diff scores were generated for the entire dataset, culminating in a total of 185 windows (i.e. 185 diff scores). Using our pre-determined threshold, 16 windows with diff scores higher than the threshold were identified as shown in Figure 6.6. A summary of the anomalous windows and their diff scores is represented in Table 6.4a.

Threshold = 0.00216024	
Anomalous windows	**Diff Score**
window 163	0.099775081
window 164	0.12644391
window 165	0.126265512
window 166	0.12266697
window 167	0.119892574
window 168	0.116689842
window 169	0.113587351
window 170	0.116997225
window 171	0.114008493
window 172	0.119197875
window 173	0.121269935
window 174	0.123674147
window 175	0.121695499
window 176	0.120863596
window 177	0.112604565
window 178	0.117216779

(a) Field Flooding attack

Threshold = 0.00216024	
Anomalous windows	**Diff Score**
window 79	0.029395
window 80	0.029166
window 81	0.029152
window 82	0.029189
window 83	0.029192
window 84	0.02324

(b) SYN Flooding attack

Threshold = 0.00216024	
Anomalous windows	**Diff Score**
window 119	0.002651
window 122	0.012518

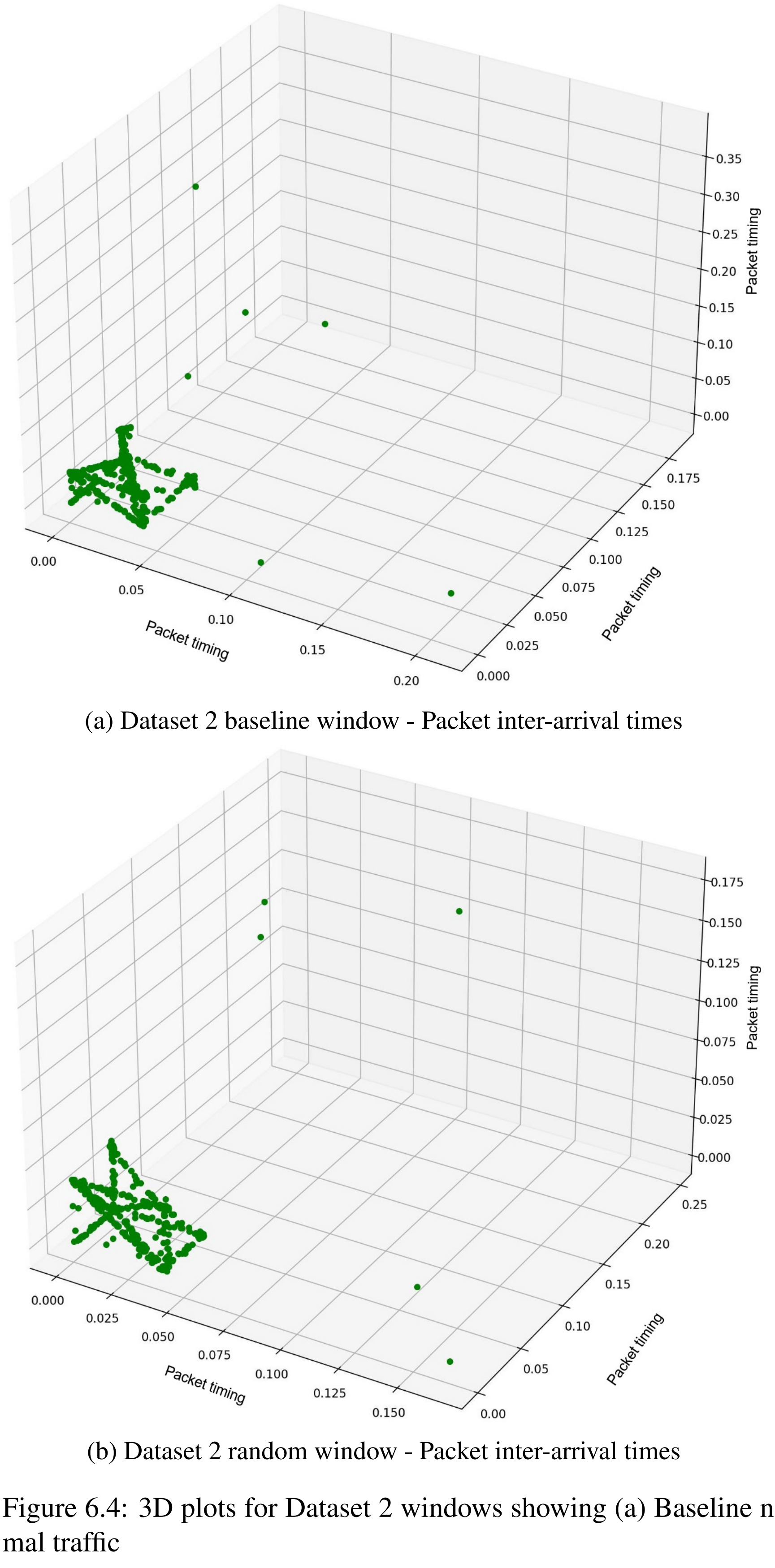
(c) MITM attack

Table 6.4: Anomalous windows with diff scores higher than set threshold (a) Dataset 2, (b) Dataset 3, (c) Dataset 4

To confirm our findings visually, a plot of any of the anomalous windows (e.g. window 163) was created (see Figure 6.5) and it showed an obvious deviation from the baseline window pattern seen in Figures 6.4a and 6.4b. Finally, to evaluate the detection of the field flooding attack using the diff score methodology, an F1 score of 0.97 was obtained.

6.5.3 Diff Score generation for Dataset 3 (SYN Flooding attack):

The same methodology was applied to Dataset 3 with the SYN flooding attack. The baseline window and random window plots are shown in Figures 6.7a and 6.7b respectively while a diff score of **0.00069579** was generated from their comparison. Again, with a



(a) Dataset 2 baseline window - Packet inter-arrival times

(b) Dataset 2 random window - Packet inter-arrival times

Figure 6.4: 3D plots for Dataset 2 windows showing (a) Baseline normal traffic, (b) normal traffic

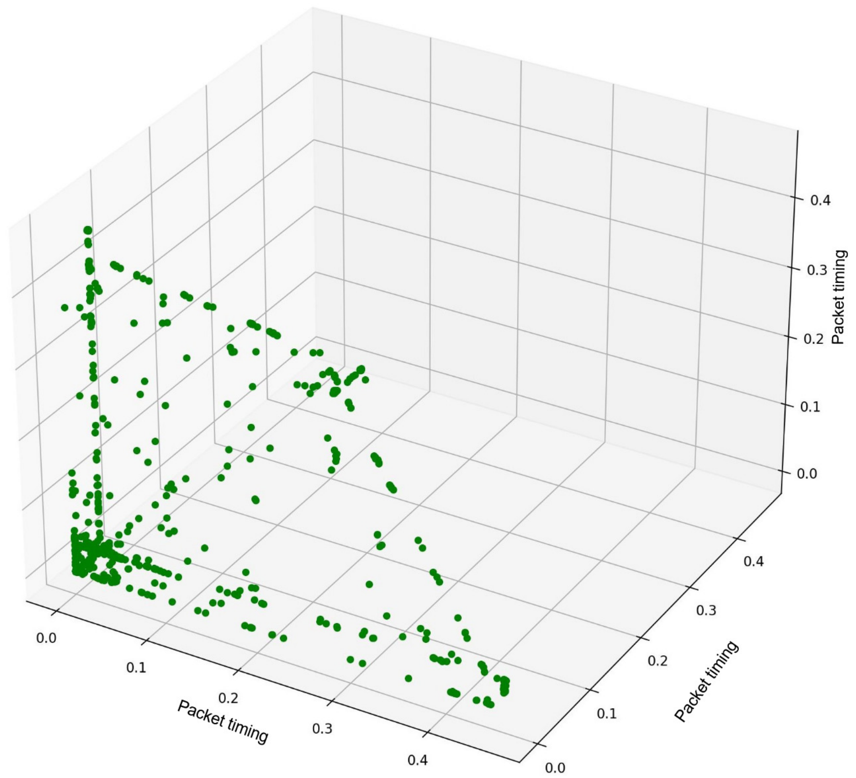


Figure 6.5: Field Flooding attack anomalous window 163 - Packet inter-arrival times

visual inspection, a similarity of patterns is observable but can only be confirmed using the diff score. The generated diff score was also below the threshold of **0.00216024** which confirms that the random window (i.e. packet index 6,000 - 7,499), when compared with the baseline window is normal traffic.

Diff scores were generated for the entire dataset, which resulted in 90 windows/diff scores. Six windows were identified to have diff scores higher than the set threshold and are summarised in Figure 6.9 and Table 6.4b.

Again, to confirm our findings visually, a plot of any of the anomalous windows (e.g. window 79) was created (see Figure 6.8) and it showed a clear lack of similarity from both Figures 6.7a and 6.7b which represent normal traffic patterns. Finally, to evaluate the detection of the syn flooding attack using the diff score methodology, an F1 score of 0.923 was obtained.

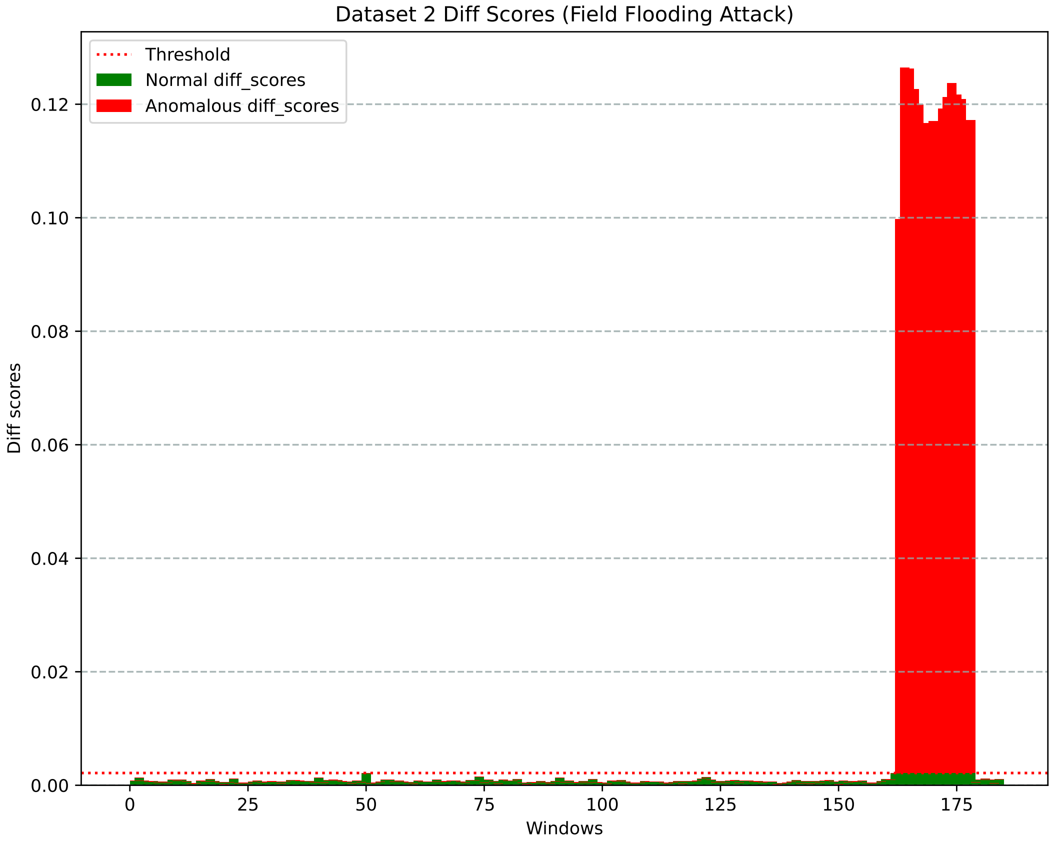


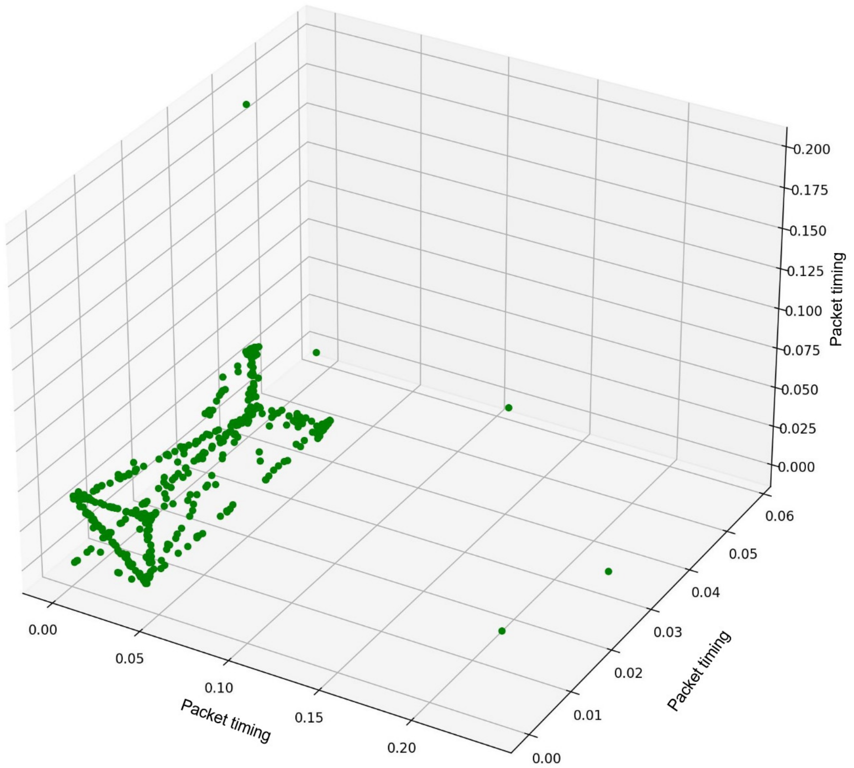
Figure 6.6: Diff Scores for Dataset 2 - Field Flooding Attack

6.5.4 Diff Score generation for Dataset 4 (MITM attack):

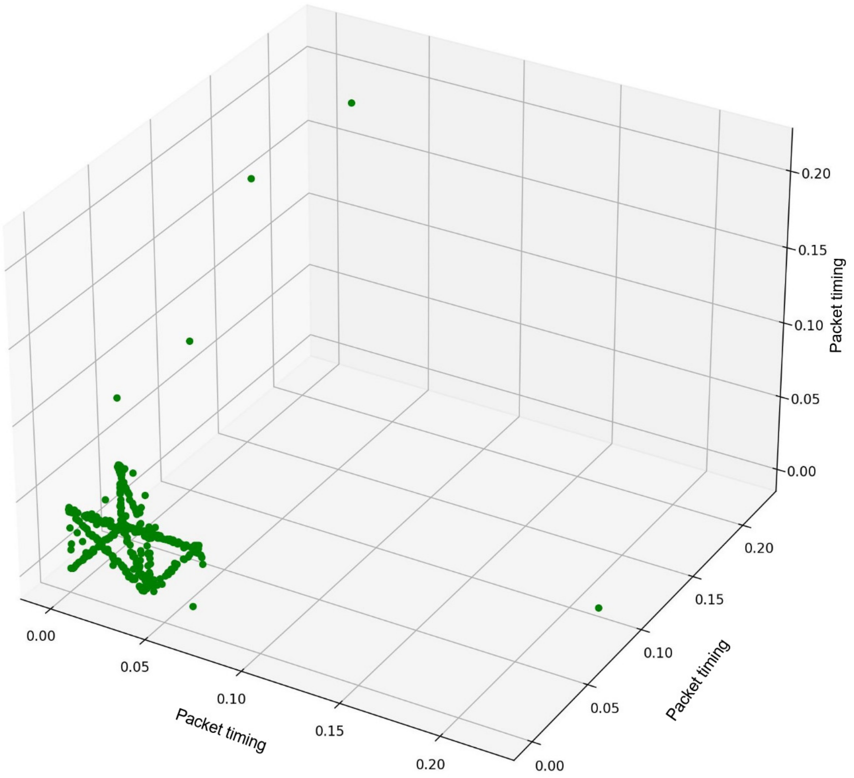
For the final dataset containing MITM attacks, the same methodology was applied to select the baseline window and random window plots. A diff score of **0.00070328** was generated from their comparison. The window selected was confirmed to be normal traffic as the diff score fell below the set threshold.

In the same manner, diff scores were generated for the entire dataset, which resulted in 125 windows/diff scores. For the MITM dataset, because it is a much stealthier attack, only two windows were identified to have diff scores higher than the set threshold and are summarised in Figure 6.12 and Table 6.4c.

Finally, to confirm our findings visually, a plot of any of the anomalous windows (e.g. window 119) Figure 6.11) was observed to be distinctively different in the pattern when



(a) Dataset 3 baseline window - Packet inter-arrival times



(b) Dataset 3 random window - Packet inter-arrival times

Figure 6.7: 3D plots for Dataset 3 windows showing (a) Baseline normal traffic, (b) normal traffic

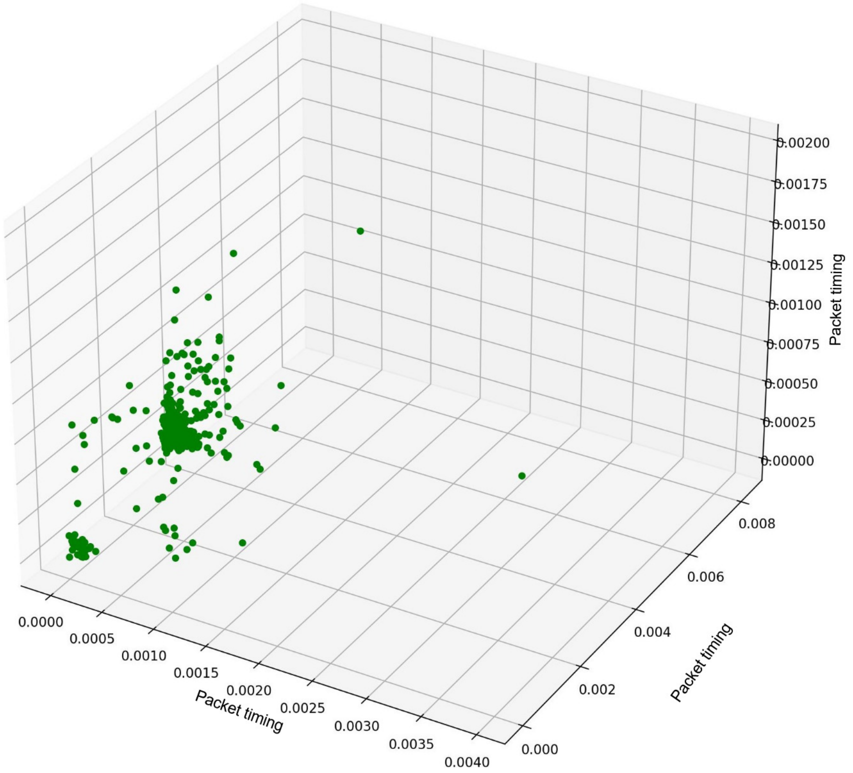


Figure 6.8: SYN Flooding attack anomalous window 79 - Packet inter-arrival times

compared both Figures 6.10a and 6.10b which represent normal traffic patterns. However, when evaluating the detection of the MITM attack using the diff score methodology, an F1 score of 0.8 was obtained. The reason for the lower F1 score when compared with the previous field flooding and SYN flooding attacks is that the MITM is a stealthier attack that is mostly detected at two points: (a) when ARP poisoning begins, and (b) when the ARP table is reverted to its original state. An interesting observation is that the ARP poisoning (the start of the MITM attack) began in window 118, but the diff score of that window is below the threshold. However, the diff score of the following window 119 was above the threshold. This may be due to the fact that the ARP poisoning packets occurred towards the tail end of window 118. However, it is still very interesting that the STADe methodology was able to observe a change in network traffic pattern in the next window 119. Lastly, it did correctly flag window 122 as anomalous, which is when the ARP table was reverted to its original state, signifying the end of the MITM attack.

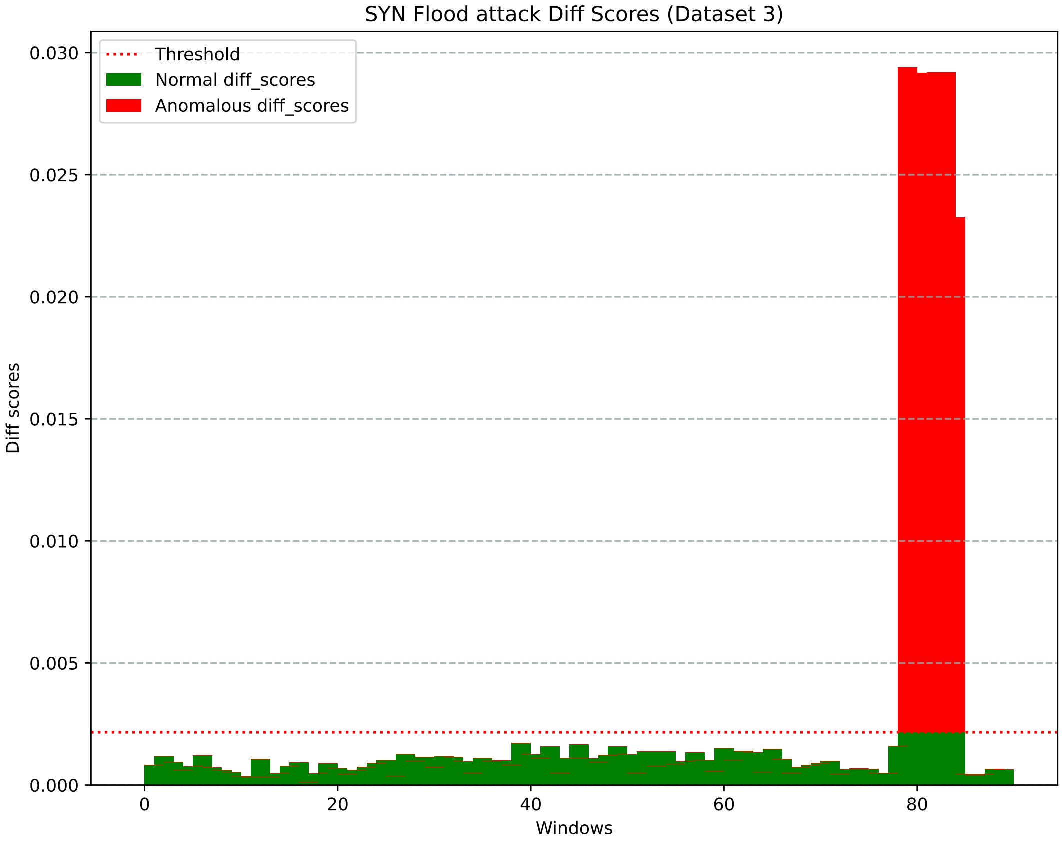
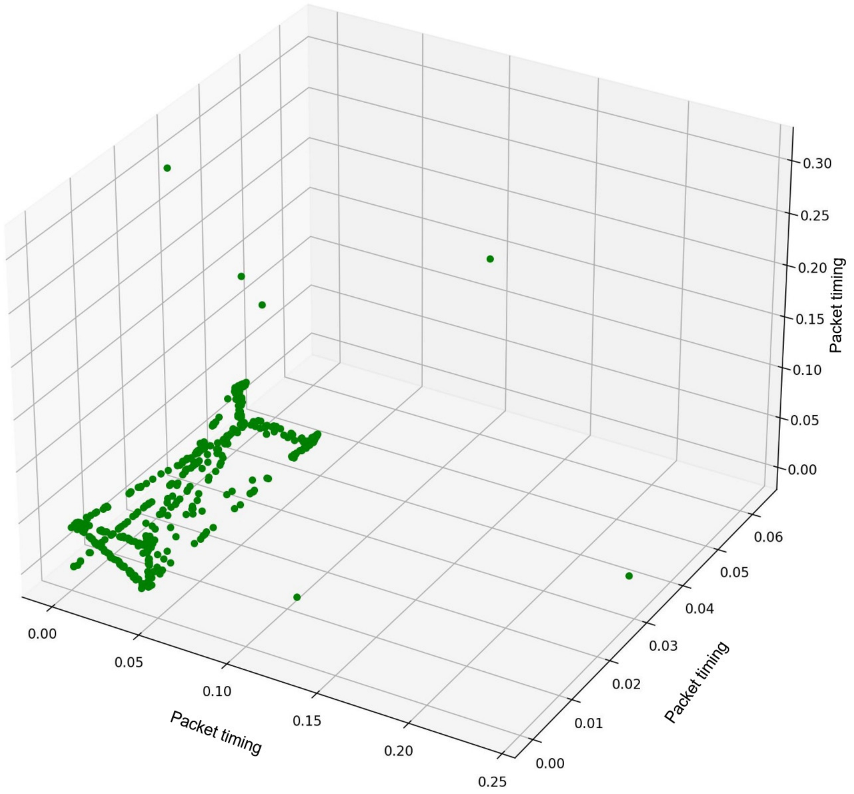


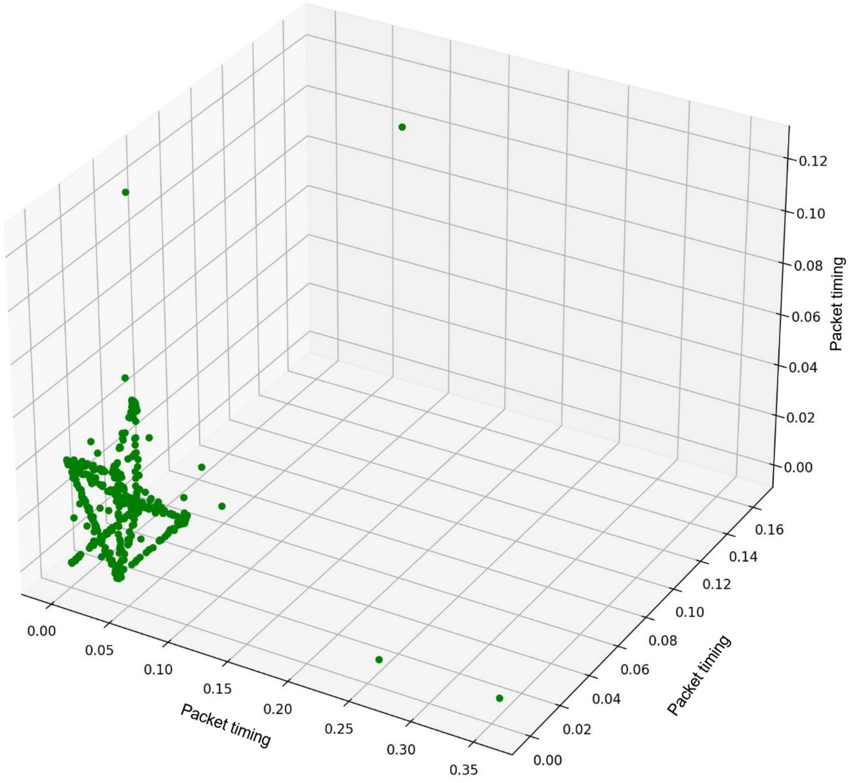
Figure 6.9: Diff Scores for Dataset 3 - SYN Flooding Attack

6.5.5 Summary of results

In summary, STADe was able to detect all the attacks by generating diff scores and having the right threshold setup for effective detection. The method proved to be effective in detection with no false positives recorded for any of the attacks. The field flooding attack had the highest F1 score of 0.97 and also had the highest number of anomalous windows (16). The reason for this is because the impact of the field flooding attack on the system lasted the longest. For the SYN flooding attack, an F1 score of 0.923 was achieved over 6 anomalous windows. Finally, for the MITM attack, the lowest F1 score of 0.8 was obtained. As explained earlier, this could be attributed to the fact that because the ARP poisoning occurred at the end of the window, the distortion was not significant enough for it to be detected as an anomaly, however, the detection occurred in the next window. One



(a) Dataset 4 baseline window - Packet inter-arrival times



(b) Dataset 4 random window - Packet inter-arrival times

Figure 6.10: 3D plots for Dataset 4 windows showing (a) Baseline normal traffic, (b) normal traffic

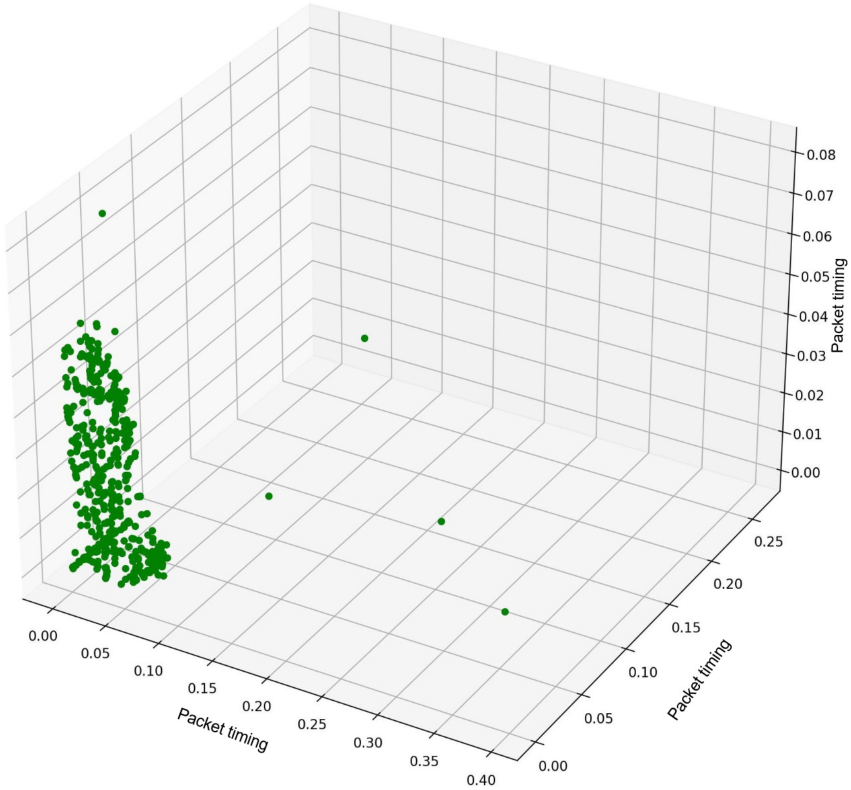


Figure 6.11: MITM attack anomalous window 119 - Packet inter-arrival times

of the most important metrics regarding anomaly detection is the False Discovery Rate (FDR), which was zero for all attacks evaluated. This means that the STADe methodology was able to effectively measure the periodicity of industrial network traffic and also, segment the traffic into equally sized windows which were further compared with each other to detect deviations from normal patterns – in essence, anomalies. Furthermore, the selection of a window size of approximately 2,500 packets (representing about one minute of network traffic) indicates that this tool can potentially detect anomalies within a minute of their occurrence. However, this would depend on the deployment of STADe in an online network monitoring and detection system. Early detection of anomalies would represent significant progress in the protection of critical infrastructure from cyber-attacks. The summary of results is highlighted in Table 6.5.

Although in Section 6.2 the limitations of unsupervised ML methods in anomaly detection were articulated, it would still be useful to compare their performances and analyse STADe with the state of the art as most anomaly detection solutions employ ML methods. Amongst these, unsupervised ML anomaly detection algorithms are most closely

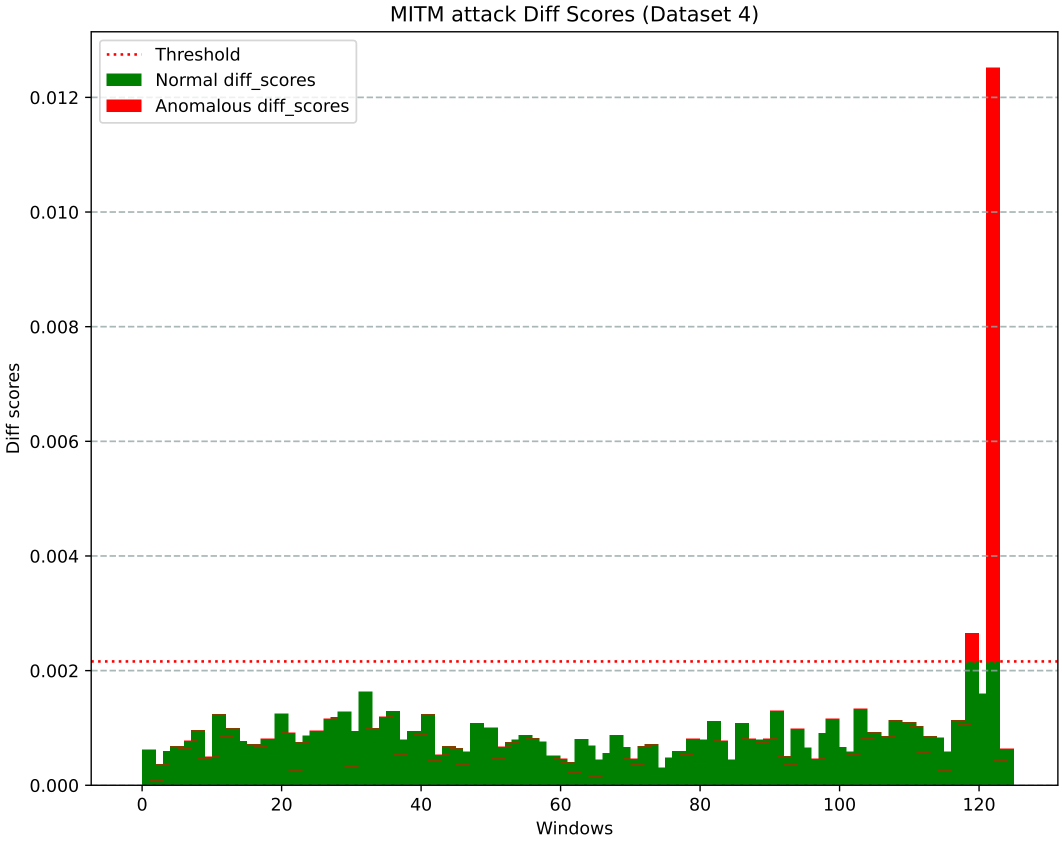


Figure 6.12: Diff Scores for Dataset 4 - MITM Attack

related and could be applicable to industrial and operational scenarios (i.e. unlabelled data). Therefore the combined version of the same dataset used in the experiments in Chapter 5 was also utilised to enable a direct comparison of STADe with the results in Chapter 5. It showed that the STADe methodology outperformed the KNN, isolation forest, and LOF algorithms in detecting anomalies in the industrial network dataset with an F1 score of 0.933 and FDR score of 0 (meaning zero false positives). Also, with respect to FDR scores, the KNN algorithm performed closest to the STADe methodology because they both utilise a distance-based algorithm that ranks each point based on the distance between the point and the nearest point, and identify the top-most points as anomalies. This is highlighted in Table 6.6.

Table 6.5: Summary of STADe performance on the datasets

Attack	**Total Windows**	**AnomalousWindows**	**Precision**	**Recall**	**F1 Score**	**FDR**	**FPR**
Field Flooding	185	16	1.0	0.94	0.97	0.0	0.0
SYN Flooding	90	6	1.0	0.86	0.923	0.0	0.0
MITM	125	2	1.0	0.67	0.8	0.0	0.0

Table 6.6: Comparing STADe results with KNN, isolation forest, and LOC algorithms (for FPR/FDR lower is better)

Algorithm	**Precision**	**Recall**	**F1-Score**	**FPR**	**FDR**
KNN	0.999	0.379	0.55	5.08 e^{-6}	0.00033
Isolation Forest	0.607	0.755	0.673	0.0196	0.3933
LOF	0.366	0.455	0.408	0.0317	0.6339
STADe	**1.0**	**0.875**	**0.933**	**0.0**	**0.0**

6.6 Potential Mitigation of O&G Attacks Using STADe

In Chapter 2, Table 2.3 highlighted some potential attacks targeting oil and gas systems. It was evident that many of the attacks analysed began with the attacker compromising the industrial network before carrying out further attacks to target various sub-systems. Following the results of STADe, Table 6.7 expands on Table 2.3 to show how STADe could potentially be used to detect the initial phases of these attacks. Further potential mitigation strategies that could be deployed after detection with STADe are also highlighted.

Furthermore, Table 6.7 indicates that attacks executed through the attack vectors identified in the case study of a subsea control system in Section 2.4.2 could potentially be detected by the STADe methodology. However, to effectively secure and mitigate these attacks against subsea control systems, detection using STADe will have to be combined with other security solutions (e.g. encryption, network segmentation, firewalls, and access control lists, e.t.c.). This would help to create a defence-in-depth approach to securing O&G systems.

It is worthy of note that when examining mitigation strategies such as encryption, it is ad-

Table 6.7: O&G attacks and mitigation using STADe

O&G Process	Attacker Motive	1st Phase Attack	Potential Further Attack	Target Component	1st phase detected by STADe?	Potential Further Mitigation Strategies
Oil tank storage	Service disruption	MITM	Oil tank level spoofing	Level sensors	Yes	Encryption, network segmentation
Hydrocarbon separation	Revenue loss	MITM	Data Tampering	Pressure or temperature sensors	Yes	Device authentication, encryption
Oil delivery, export, piping	Service disruption	MITM	Command injection	PLC, pumps, actuators	Yes	Device authentication, encryption
Emergency shutdown	Damage to asset	DoS / DDoS	DoS	Safety Instrumented System, PLC, and actuators	Yes	Firewalls, access control lists, Packet rate limiting
Custody transfer-/metering	Revenue loss, theft of operational information	MITM	Wellhead production data exfiltration	Flow computers, meters, pressure or temperature sensors	Yes	Network segmentation, access control
Subsea production	Damage to asset, loss of production	MITM	Choke size replay attack	PLC, actuators	Yes	Encryption, access control, digital certificates
Subsea production	Theft of operational information	MITM	Interception of commands and sensor readings	Master control station	Yes	Network segmentation, access control
Subsea production	Damage to asset, potential loss of lives	MITM	Injecting falsified sensor data	Sensors and PLC	Yes	Encryption, access control, digital certificates

vised to consider its potential impact on system performance and communication latency through further analyses. This is especially important for safety-critical processes and is further emphasised in several industry standards, guidelines, and recommended practices such as NIST SP 800-82 [9], ISA/IEC 62443-3-3:2013 System Security Requirements and Security Levels, API Standard 1164 - Pipeline SCADA Security, and Oil & Gas UK Cyber Security Framework for Oil & Gas, 2017.

6.7 Conclusions

Anomaly detection in industrial networks has had a problem with high false positive rates and high computational complexities which has hindered its widespread use in practice. The aim of this Chapter was to improve upon the state of the art unsupervised anomaly detection methods (mostly ML-based) by reducing the limitations high false alerts together with a low detection rate. The novel STADe methodology introduced in this chapter represents an unsupervised time-window-based approach to anomaly detection in industrial control networks that results in zero false positives. This chapter focused on a computationally efficient mechanism to detect a break in periodicity, which flags anomalies. For this reason, a single feature of packet inter-arrival times was recorded as point events. The aim was to characterise the periodicity of any given industrial network using the packet timings to create a mini-model of the system representing the normal operation pattern. This mini-model is represented as the baseline window, which further acts as the sliding window that is used to compare with the rest of the traffic windows.

There are two key differences in the approach employed in experiments carried out in this chapter from our previous experiments carried out in Chapters 4 and 5 which can be summarised as:

- The transition from packet-based analysis to window-based analysis. This resulted in a significantly reduced number of false alerts.

- The reduction in number of features extracted from data stream for analysis. This reduced the computational complexity of the analysis.

The results from the experiments showed no false positives with F1 scores of 0.97, 0.923, and 0.8 recorded for the detection of field flooding, SYN flooding, and MITM attacks respectively. In order to assess the performance of STADe in relation to other unsupervised machine learning algorithms investigated in Chapter 5, namely K-Nearest Neighbors (KNN), Isolation Forest, and Local Outlier Factor (LOF), the STADe methodology was applied to the combined dataset used in Chapter 5. Notably, STADe demonstrated superior performance, achieving an F1 score of 0.933 when applied to the same dataset in comparison to F1 scores of 0.55, 0.673, and 0.408 for KNN, Isolation Forest, and LOF, respectively. More importantly, STADe recorded zero false positives (0% FDR) compared to the FDR scores of 39.33%, 63.39%, and 0.03% recorded by isolation forest, LOC, and KNN algorithms respectively. This makes STADe very promising to explore further.

Essentially, from the experiments, the STADe methodology was able to:

- measure the periodicity of a given industrial network in the form of a pattern,
- segment the network traffic into time windows which were further compared with each other to detect deviations from the normal pattern established in order to detect anomalies, and
- as an additional step, map this pattern onto a 3-dimensional space visually, if required.

The fact that it utilises a single feature of packet timings that is unaffected by network encryption means it could potentially be integrated with other security solutions simultaneously to improve the security posture of industrial networks.

One important application in the real world for STADe is its potential usefulness if used in conjunction with a human-in-the-loop to narrow down large volumes of data and enable quick identification of anomalous packets within a time window and investigate further to determine the cause of the anomaly. For our specific test case in this chapter, with a window size of approximately 2,500 packets (representing 1 minute of network traffic), it means that an attack can potentially be detected within a minute of its occurrence. However, for this to be achieved, STADe would need to be deployed within an online network monitoring and detection setup. This could also potentially put an end to scenarios where attacks carried out are undetected for several months.

Furthermore, this Chapter demonstrated the ability of STADe to detect attacks that could be executed through attack vectors identified in Chapter 2 targeting some offshore systems and subsea control systems. If deployed effectively alongside other highlighted mitigation strategies, STADe can be utilised to significantly improve the security of O&G systems in a defence-in-depth approach.

Finally, this chapter has shown that the periodicity of industrial network communications can be measured and deviations from this periodicity could be determined in an unsupervised manner to develop an anomaly detection model capable of achieving high detection accuracy. Also, in light of the notably low FDR score observed by the distance-based method in Chapter 5, we integrated the L2-norm (Euclidean distance) into the novel STADe methodology. Upon its application to the datasets, STADe yielded a commendable outcome with a FDR/FPR of zero. These are early promising results that require further evaluation. In the next chapter, this methodology shall be further evaluated on a public industrial dataset to assess its ability to consistently record low FDR/FPR scores with high detection rates in a different industrial network. In other words, is STADe adaptable across different industrial networks? This leads us to our sixth research question which would be investigated in Chapter 7.

Chapter 7

Testing Robustness and Adaptability of STADe Performance Results on Public Datasets

7.1 Introduction

In Chapter 6, we introduced STADe, a novel methodology that uses the time periodicity behaviour associated with industrial networks to detect anomalies in network transmission. This was possible by first extracting the packet inter-arrival times from an industrial network data stream, then slicing the extracted data into time windows, and using a sliding time window (representing regular traffic) to compare with other windows and generating a diff score to determine if the window under consideration is anomalous or not. The datasets used to evaluate the STADe methodology so far were generated from a gas wellhead monitoring testbed.

The purpose of this chapter is to further evaluate the STADe methodology and its performance using a publicly available industrial dataset. This will determine if the results obtained on the gas wellhead monitoring testbed are consistent on a different industrial network with varying devices, equipment, and network architecture. This will also provide answers to the sixth and final research question:

RQ6 *Is the investigated unsupervised anomaly detection method adaptable across differ-*

ent industrial networks?

In answering this question, this chapter will provide the eighth contribution of this research:

C8 *This research determines the adaptability of the STADe methodology described in chapter 6 to different industrial cyber-physical networks.*

7.2 Method

For this chapter, the method employed remains the same as in chapter 6 as this is a further evaluation of the STADe methodology and its performance in a different industrial network. By using the same methods and keeping the hyperparameter options similar (i.e. window size, diff weight = 0.5) - except for the threshold value - it is possible to evaluate the performance and compare because the conditions have been kept as close as reasonably practicable.

The experiments carried out in this chapter are identical to those carried out in Chapter 6. As before, we will utilise four datasets (1 normal traffic, 3 attacks) for our experiments using similar conditions. The only exception will be the determination of a threshold value - which is unique for every industrial network. For each dataset, the first window will be designated as the sliding window containing normal traffic, which will then be compared to the rest of the windows by generating diff scores. Recall that, to understand the diff scores, the closer the value is to zero, the more similar the window is compared to the sliding window and vice versa. Therefore, since the first window is the sliding window, the diff score for *Window 0* will always be zero as this is a comparison of the sliding window with itself (identical traffic pattern).

In the following section, the dataset selection process, attack description, and the process of determining the threshold for our experiments are explained.

7.3 Datasets and Threshold Determination

7.3.1 Dataset Selection

The next task is to select a suitable industrial network dataset that would be used to evaluate STADe's performance. Recall that the challenge with general industrial dataset availability for security research was highlighted in Chapter 2. As the requirement in this chapter is more specific for industrial network data, thirteen publicly available datasets were explored for their suitability [224]. For the dataset to be considered fit-for-purpose, it is generally expected to:

- be from an industrial network with multiple PLCs and sensors/actuators. The more devices on a testbed, the closer its data is to real-world data because real OT environments are highly dynamic in nature;
- have a pcap file format;
- have at least one dedicated pcap file with only normal traffic (no attack) to enable determination of a suitable threshold, and
- at least contain popularly executed attacks compromising the availability of devices in industrial environments (e.g. DoS attacks). Attacks against integrity of the data are also desirable, however, as a minimum, the dataset must have DoS attacks. This is because, in critical infrastructure environments, availability has the highest priority compared to integrity and confidentiality as they are usually deployed to control critical services that should be in continuous operation (e.g. traffic lights, electrical grids, or gas transmission lines).

Table 7.1 shows the summary of the datasets and their suitability. Only the Modbus SCADA dataset [225] meets all the criteria for our experiments so this was selected. The next sub-section describes details of this dataset and why it is suitable for our purpose.

Table 7.1: Downloadable ICS datasets from industrial networks

Dataset	Industrial Network	Pcap file Format	DoS Attacks	Dedicated normal traffic pcap
4SICS [226]	●	●		
Cybercity Dataset [227]	●	●	●	
D2: Gas Pipeline [146]	●			
D3b: Water S. T. [146]	●		●	
D4: New Gas Pipeline [146]	●			
Electra Modbus [228]	●			
Electra S7Comm [228]	●			
HVAC Traces [229]	●	●		
Lemay SCADA [230]	●	●		
Modbus SCADA [225]	●	●	●	●
S4x15 ICS [231]	●	●		
WUSTL-IIOT-2018 [87]	●			
SWaT [232]	●	●		

7.3.2 Dataset Description

Testbed Setup

The chosen dataset was generated on a small-scale process automation scenario using ModbusTCP protocol to emulate a cyber-physical system [225]. The testbed (shown in Figure 7.1) consists of a liquid pump simulated by a 3-phase electric motor controlled by a variable frequency drive (VFD) that allows for multiple rotor speeds. This VFD is controlled by a PLC that is polled for data from the HMI. The motor speed is determined by a set of predefined liquid temperature thresholds, whose measurement is provided by a Modbus RTU fitted with a temperature gauge (simulated by a potentiometer connected to an arduino).

This testbed setup satisfies our criteria of multiple PLCs and sensors/actuators, which makes it more realistic than just a simple device being polled for data. It is also useful that the process being emulated is not limited to an oil and gas setup. This gives us a chance to generalise STADe's performance on other cyber-physical systems obtainable across other critical infrastructure industries (e.g. manufacturing, food processing, or transportation).

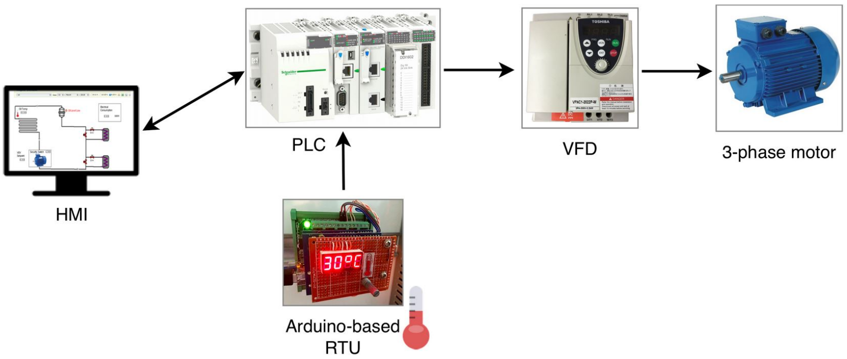


Figure 7.1: Testbed setup for Modbus SCADA dataset [225]

Attacks captured in dataset

The dataset contains three different types of DoS attacks described as follows:

- Modbus Query Flooding attack: This attack attempts to flood the PLC with read holding registers (Modbus FC 0x03) commands which may lead to side effects such as device resource exhaustion, scan cycle latency deviations, or loss of connectivity [225].

- SYN Flooding attack: This is a popular DoS attack whose primary aim is to overwhelm the capacity of the network or the networking subsystem in the target device with SYN requests.

- Ping Flooding attack: Similar to SYN flooding, this attack is also an attempt to overwhelm the capacity of the network by flooding the network with high-rate ping requests targeted at a device.

Table 7.2 shows a summary of the dataset files and their general description.

7.3.3 Determining the threshold

The threshold hyperparameter is a value that is used to categorise a diff score as normal or anomalous. It is unique to every industrial network and is determined from normal traffic. Similar to the method employed in Chapter 6, using the normal traffic pcap file in our dataset, diff scores were generated and the maximum score generated was selected as the threshold. The generated diff scores - from 28 windows - are shown in Table

Table 7.2: Summary of datasets used

Dataset type	**File Name**	**Attack Duration**	**Capture Duration**	**Total No. of Pkts**
Normal traffic	eth2dump-clean-1h_1	N/A	1 hr	72150
Query flooding dataset	eth2dump-modbusQuery2Flooding1m-1h_1	1 min	1 hr	106913
SYN flooding dataset	eth2dump-tcpSYNFloodDDoS1m-0-5h_1	1 min	30 mins	45271
Ping flooding dataset	eth2dump-pingFloodDDoS1m-1h_1	1 min	1 hr	47387

7.3 which indicates that Window 18, with a diff score of **0.004928039** has the maximum value and is therefore used as the threshold. These scores are also visualised in the barplot shown in Figure 7.2. For our subsequent experiments, any diff score generated higher than **0.004928039** will be flagged as an anomalous window.

7.3.4 Evaluation Metrics

To ensure a uniform comparison of the performance of STADe, we shall be using the same evaluation metrics described in Chapter 6 (i.e. precision, recall, F1 score, False Discovery Rate (FDR), and False Positive Rate (FPR)). This entails labelling the datasets, which is feasible because the attack start and end times are known from the dataset description.

The results of these experiments are discussed in the next section.

7.4 Results

The experiments carried out include generating diff scores for the remaining 3 datasets containing attacks (i.e. query flooding, SYN flooding, and ping flooding attacks) and using our threshold value of **0.004928039** to classify the windows as normal or anomalous. The first window in each dataset will be selected as the sliding window. In addition, the sliding window is visualised in 3-dimensional space, alongside another window contain-

Table 7.3: Diff scores for eth2dump-clean-1h_1 pcap - The highest score of 0.004928039 is selected as threshold

Window Num	Diff score	Window Num	Diff score
Window 0	0	Window 14	0.000874597
Window 1	0.002083958	Window 15	0.00416434
Window 2	0.002873405	Window 16	0.001952591
Window 3	0.003095275	Window 17	0.000946216
Window 4	0.003184019	**Window 18**	**0.004928039**
Window 5	0.002843107	Window 19	0.003188699
Window 6	0.002977615	Window 20	0.002460945
Window 7	0.001686727	Window 21	0.002027755
Window 8	0.002623678	Window 22	0.002315847
Window 9	0.001930151	Window 23	0.003139994
Window 10	0.002001565	Window 24	0.002797589
Window 11	0.00251402	Window 25	0.002133557
Window 12	0.002514642	Window 26	0.001106057
Window 13	0.000684406	Window 27	0.001277524

ing normal traffic. While this helps the user to visualise the normal traffic pattern and its slight variations, it can not be used to determine anomalous windows. For that, the numerical diff score is used to identify a significant variation in window traffic. Finally, as an additional step, any identified anomalous window is visualised in 3-dimensional space to show if there are any significant changes in the previously visualised pattern.

The results of using STADe on each attack dataset are discussed in subsequent subsections.

7.4.1 Diff score generation for query flooding attacks

For this experiment, the dataset containing query flood attacks was used (filename: *eth2dump-modbusQuery2Flooding1m-1h_1*) and, as described previously, the first window was selected as the sliding window to compare with the rest of the windows to generate the scores. Next, diff scores were generated for the entire dataset which resulted in 41 windows according to our window size (see Table 7.4a). Out of these, 16 windows (Windows 2 - 17) had diff scores higher than the specified threshold and were flagged as anomalous. Figure 7.5 shows the distribution of the diff score values highlighting the anomalous windows in red. The pattern shown in Figure 7.5 suggests that the query flooding attack was executed near the start of the data collection.

Threshold = 0.004928039			
Query Flood Attack			
Window Num	**Diff score**	**Window Num**	**Diff score**
Window 0	0	Window 21	0.002323
Window 1	0.004449038	Window 22	0.002981
Window 2	**0.054189135**	Window 23	0.004593
Window 3	**0.073695999**	Window 24	0.00388
Window 4	**0.076536555**	Window 25	0.002796
Window 5	**0.074771973**	Window 26	0.002051
Window 6	**0.079117958**	Window 27	0.001472
Window 7	**0.074505857**	Window 28	0.003538
Window 8	**0.075232847**	Window 29	0.003155
Window 9	**0.076022422**	Window 30	0.001595
Window 10	**0.078053148**	Window 31	0.003034
Window 11	**0.075421144**	Window 32	0.002746
Window 12	**0.076650883**	Window 33	0.003089
Window 13	**0.078800008**	Window 34	0.003063
Window 14	**0.07743527**	Window 35	0.004518
Window 15	**0.077793741**	Window 36	0.003797
Window 16	**0.07771486**	Window 37	0.003881
Window 17	**0.074762766**	Window 38	0.003717
Window 18	0.003829012	Window 39	0.003368
Window 19	0.002984399	Window 40	0.003126
Window 20	0.00279772		

(a) Diff scores for query flooding pcap

Threshold = 0.004928039			
SYN Flood Attack		**Ping Flood Attack**	
Window Num	**Diff score**	**Window Num**	**Diff score**
Window 0	0	Window 0	0
Window 1	0.001499606	**Window 1**	**0.155834302**
Window 2	**0.04837504**	**Window 2**	**0.225940616**
Window 3	**0.074190828**	**Window 3**	**0.22583687**
Window 4	**0.073857598**	**Window 4**	**0.225326169**
Window 5	**0.073878528**	**Window 5**	**0.224423038**
Window 6	**0.073929547**	**Window 6**	**0.020845698**
Window 7	**0.009058085**	Window 7	0.000302833
Window 8	0.00124635	Window 8	0.001766768
Window 9	0.001312027	Window 9	0.001468691
Window 10	0.002143526	Window 10	0.001849506
Window 11	0.003951452	Window 11	0.000799092
Window 12	0.004107144	Window 12	0.001073293
Window 13	0.002286754	Window 13	0.002177379
Window 14	0.00265846	Window 14	0.001110221
Window 15	0.003458969	Window 15	0.00227533
Window 16	0.002003811	Window 16	0.002530714
		Window 17	0.002231806

(b) Diff scores for SYN flooding and ping flooding pcaps

Table 7.4: Diff scores showing windows higher than set threshold (a) Query flood (Windows 2 - 17), (b) SYN flood (Windows 2 - 7), Ping flood (Windows 1 - 6)

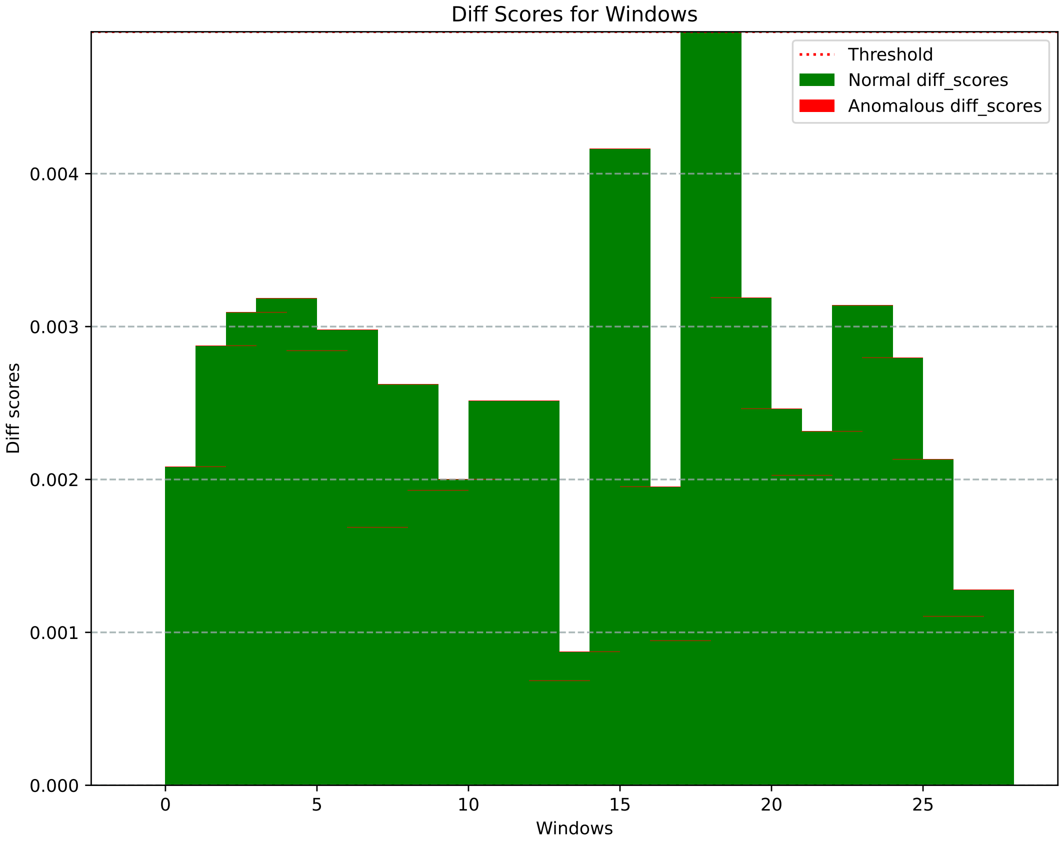
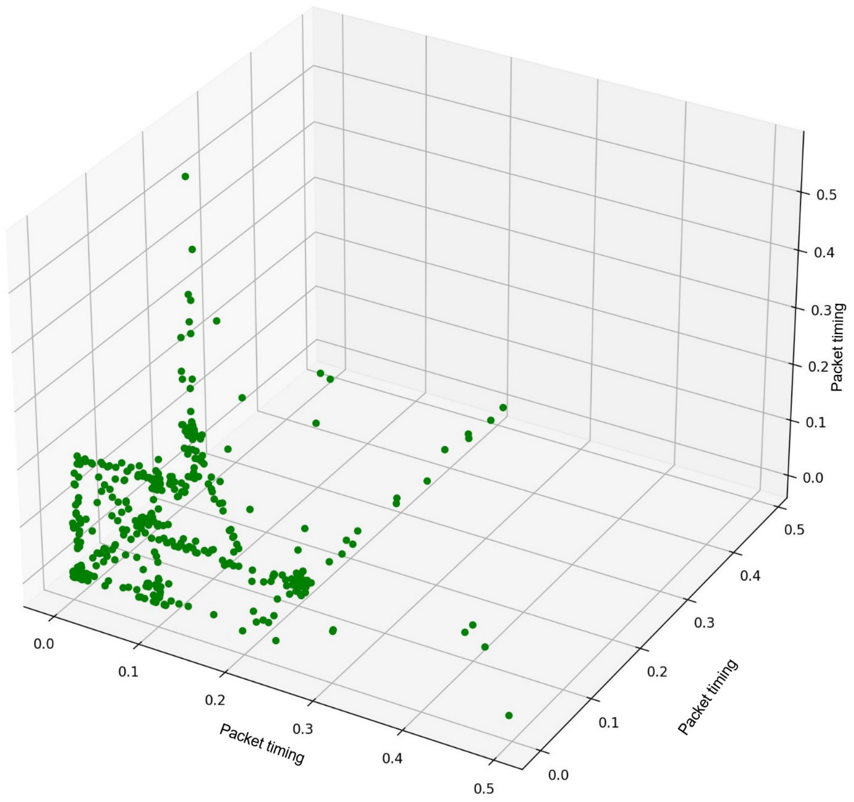


Figure 7.2: Diff Scores for normal traffic showing Window 18 having the highest value

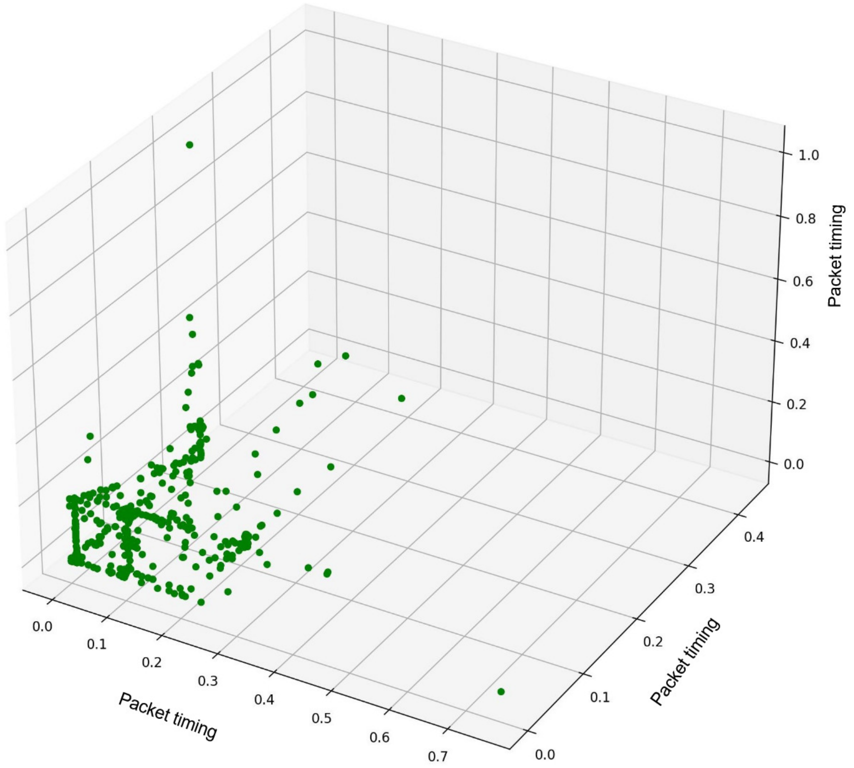
To confirm our findings visually, it is useful to view what the normal traffic pattern in this industrial network looks like. To do this, a scatter plot of the sliding window and a random window containing normal traffic was plotted in 3-dimensional space. The pattern in both these windows can be seen to be visually similar as shown in Figures 7.3a and 7.3b. Next, one of the anomalous windows - Window 5 - was also plotted in 3-dimensional space (see Figure 7.4). When compared with 7.3a and 7.3b, you can see an obvious significant change in pattern. This proves that our visual observation is consistent with the diff score results. Finally, an F1 score of 1.0 and FDR/FPR of 0 was obtained.

7.4.2 SYN Flooding Attacks

For the next dataset containing SYN flooding attacks (filename: *eth2dump-tcpSYNFloodDDoS1m-0-5h_1*), a total of 17 windows were obtained and diff scores were generated to compare



(a) Sliding window - Packet inter-arrival times



(b) Normal traffic window - Packet inter-arrival times

Figure 7.3: Visualising normal traffic windows showing (a) Sliding window, (b) Random normal traffic (for comparison)

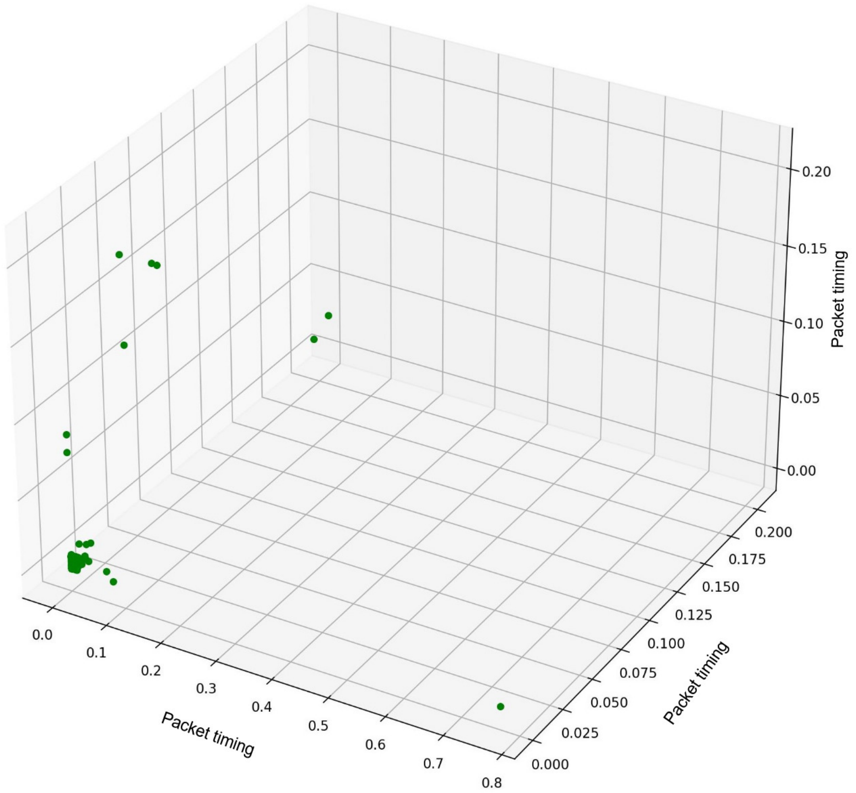


Figure 7.4: Query Flood attack anomalous window - Packet inter-arrival times

with the sliding window (first window). From these 17 windows, 6 of them (Windows 2 - 7) had diff scores higher than the threshold of **0.004928039** and thus, were identified as anomalous windows. These scores are highlighted in Table 7.4b while Figure 7.8 also shows the diff score distribution for the entire dataset indicating where the SYN flooding attack was executed.

To confirm the findings visually, the 3-dimensional scatter plots for the sliding window and random normal traffic window, again, indicate a similar pattern (see Figures 7.6a and 7.6b. However, when compared to the scatter plot for one of the anomalous windows (i.e. Window 5 - Figure 7.7), there is an obvious significant shift in pattern when compared to Figures 7.6a and 7.6b. This again shows that our visual inspection is consistent with the generated diff scores. This experiment had an F1 score of 1.0 and FDR/FPR of 0.

7.4.3 Ping Flooding Attacks

The final attack dataset in our experiment contained ping flooding attacks in the pcap file (filename: *eth2dump-pingFloodDDoS1m-1h_1*). This dataset generated 18 windows with

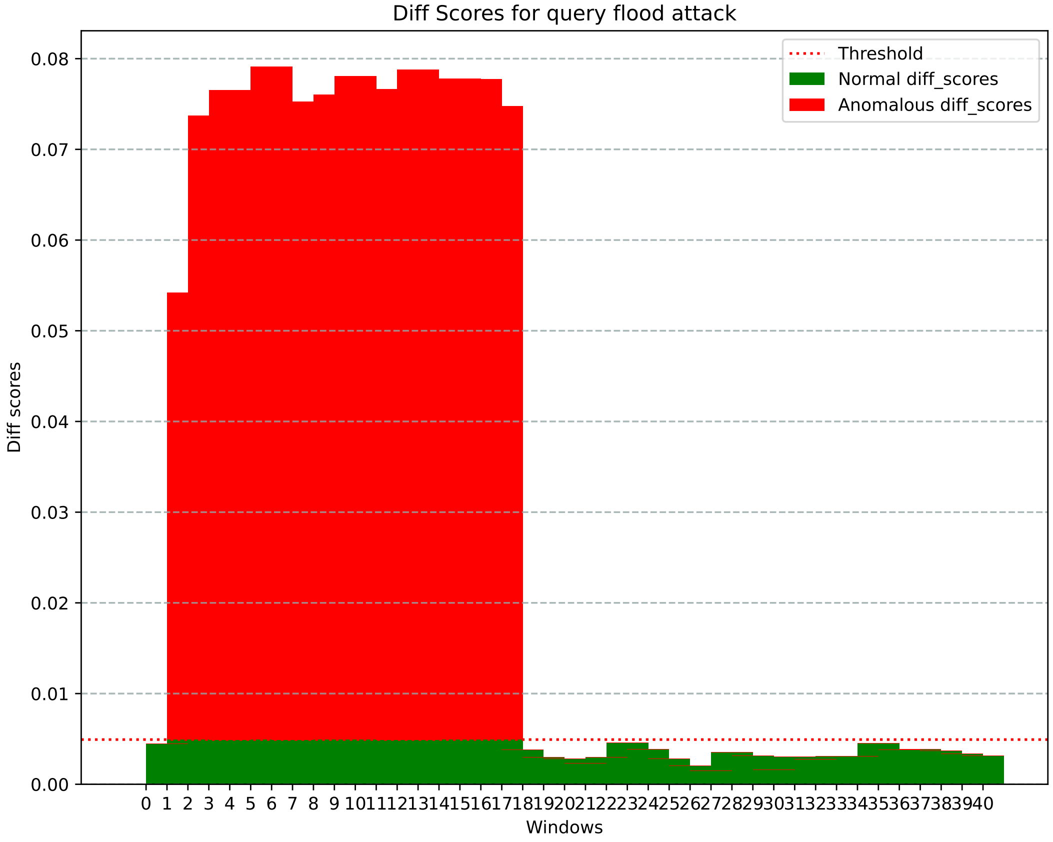
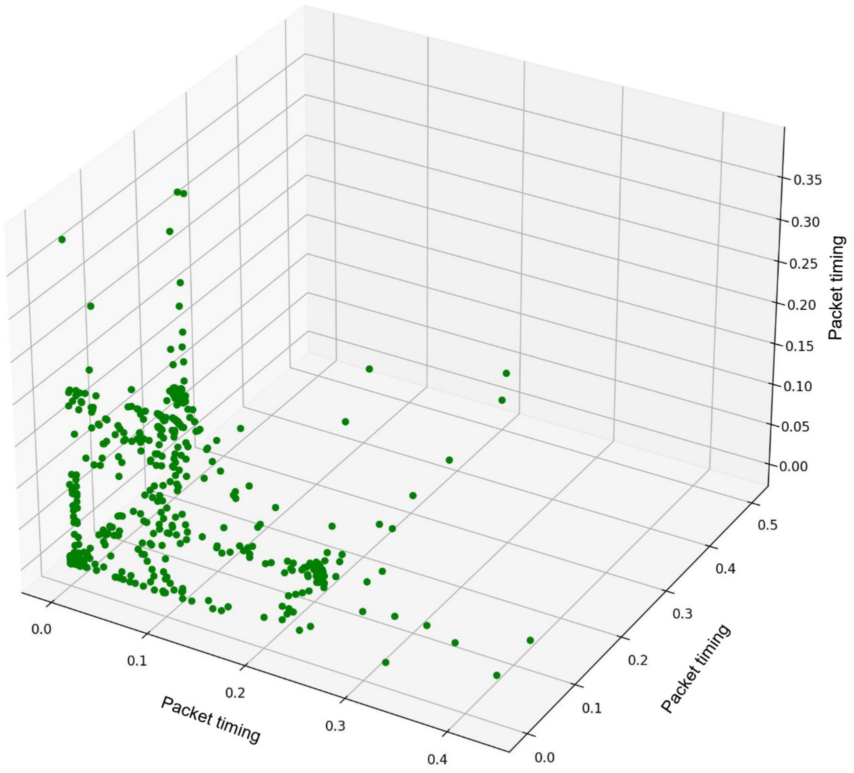


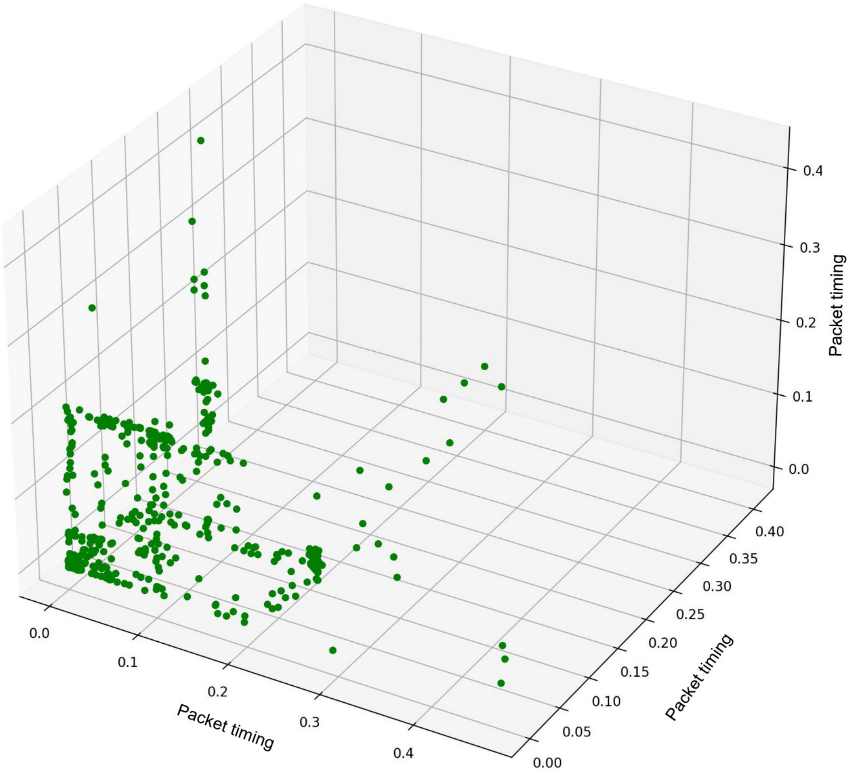
Figure 7.5: Diff Scores for Query Flooding Attack Dataset

the same corresponding number of diff scores after comparison with the sliding window. Amongst these 18 windows, 6 windows were observed to have diff scores higher than the threshold of **0.004928039** (Windows 1 - 6). The diff scores are highlighted in Table 7.4b while Figure 7.11 shows the distribution of the scores across the entire dataset with the anomalous windows highlighted in red. From Figure 7.11, it can also be observed that the ping flooding attack was executed very close to the start of the data capture.

To visualise the results for human observation, the scatter plots for the sliding window and another window containing normal traffic are shown in 3-dimensional space in Figures 7.9a and 7.9b which shows a similarity in the pattern. When compared with a 3-dimensional scatter plot for one of the anomalous windows (i.e. Window 2 - Figure 7.10), again, an obvious significant shift in pattern is observable. This shows consistency



(a) Sliding window - Packet inter-arrival times



(b) Normal traffic window - Packet inter-arrival times

Figure 7.6: Visualising normal traffic windows showing (a) Sliding window, (b) Random normal traffic (for comparison)

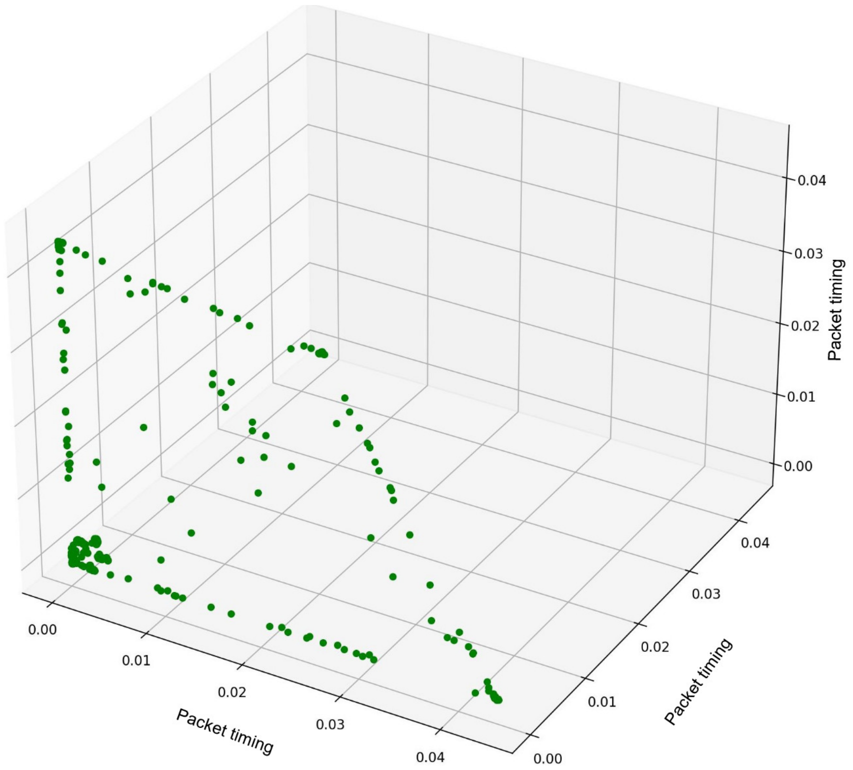


Figure 7.7: SYN Flood attack anomalous window - Packet inter-arrival times

between humans visualising the patterns and the mathematically generated diff score. Finally, this experiment had an F1 score of 1.0 and FDR/FPR of 0.

7.4.4 Summary of results

The results of the experiments carried out showed that the STADe methodology is able to detect all three attacks - query flooding, SYN flooding, and ping flooding attacks - effectively using pcap files captured from an industrial network. The F1 scores and the zero FDR/FPR obtained for all the attacks suggests that it is consistent with the results in Chapter 6. The confusion matrix for the attack detection for all three datasets (Figure 7.12) confirms how STADe classified the windows showing no false positives or negatives.

An interesting observation when examining the normal traffic pattern for the ping flooding attack in Figures 7.9a and 7.9b is that it differs from the previously observed normal traffic pattern in Figures 7.3a, 7.3b, 7.6a, and 7.6b. The reason for this can be found in the explanation given by the authors of the dataset about the testbed setup. Due to the horizontal communications between the PLC and the Modbus RTU, there is often a flooding

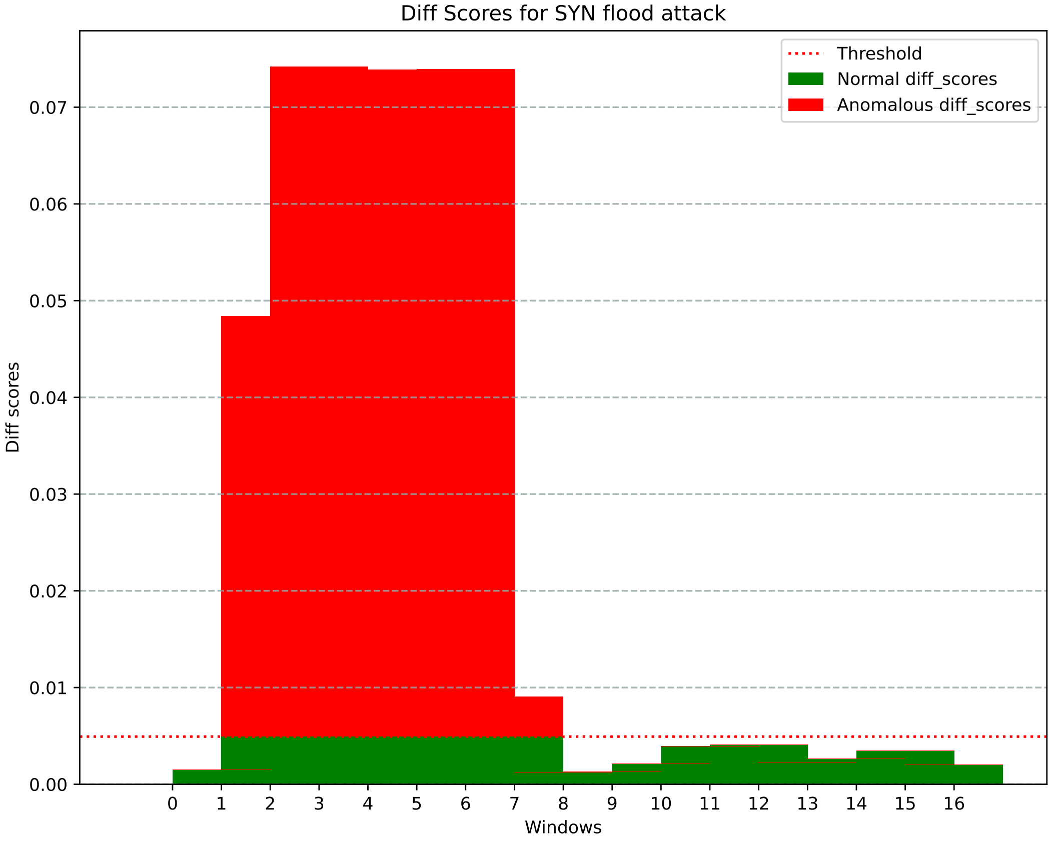
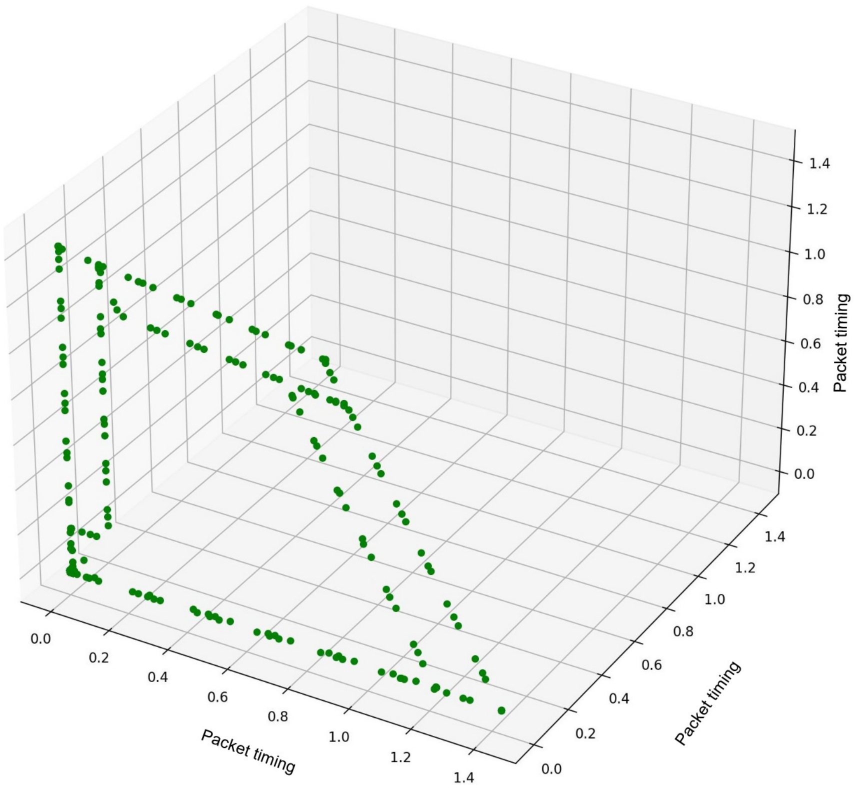
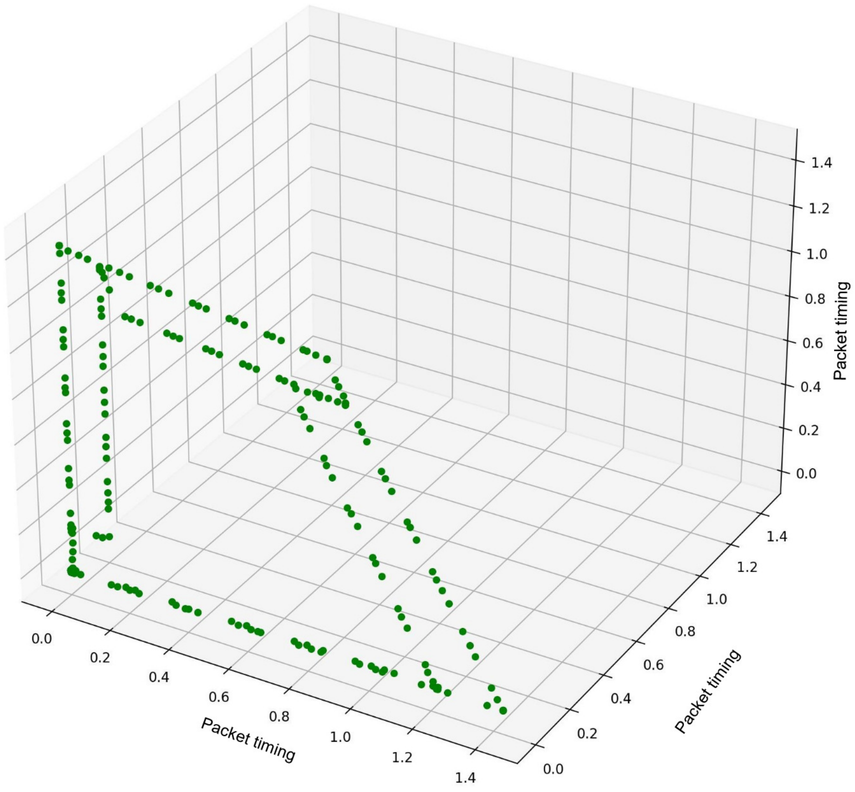


Figure 7.8: Diff Scores for SYN Flooding Attack Dataset

of PSH, ACK packets in their transmissions. The file *eth2dump-pingFloodDDoS1m-1h_1* used in the experiment had a significant portion of the PSH, ACK transmission compared to the other datasets. This is what resulted in a change in the normal traffic pattern observed in Figures 7.9a and 7.9b. This could be classed as an example of concept drift in an industrial network. In this case, this provides evidence that the STADe methodology can be utilised to create a model that could potentially mitigate the effects of concept drift without significant re-training effort, unlike other anomaly detection methods. Once the sliding window is updated to the new pattern, STADe will continue to detect anomalous windows effectively. This is a significant advantage over machine learning anomaly detection methods that generally rely on 1) training and re-training to mitigate against concept drift, and 2) large datasets to carry out the training of the models.



(a) Sliding window - Packet inter-arrival times



(b) Normal traffic window - Packet inter-arrival times

Figure 7.9: Visualising normal traffic windows showing (a) Sliding window, (b) Random normal traffic (for comparison)

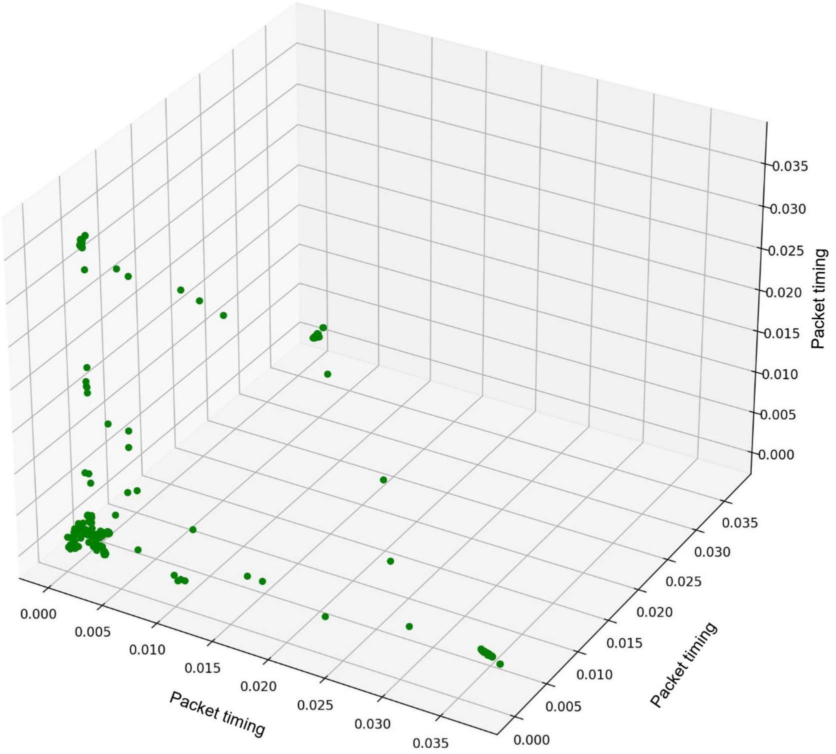


Figure 7.10: Ping Flood attack anomalous window - Packet inter-arrival times

7.5 Conclusions

This chapter set out to determine the adaptability of the STADe methodology to different industrial networks and to evaluate its effectiveness in detecting anomalies in such a network. The aim was to investigate if the results obtainable in an industrial network with varying equipment and network architecture would be consistent with the performance in chapter 6.

The datasets used were generated from a small-scale testbed emulating a process automation scenario containing three different DoS attacks a) query flooding, b) SYN flooding, and c) ping flooding. Overall, the results of the experiments showed that every window containing attacks in the datasets was correctly detected - as evidenced by the F-1 scores of 1.0 and zero false discovery rates obtained. The significance of these results lies in the fact that across two separate industrial networks representing different critical infrastructure industries (i.e. oil & gas and manufacturing), false positive and false discovery rates remained zero. This is critical because high FDR and FPR is considered one of the main downsides to the adoption of anomaly detection in real industrial environments because

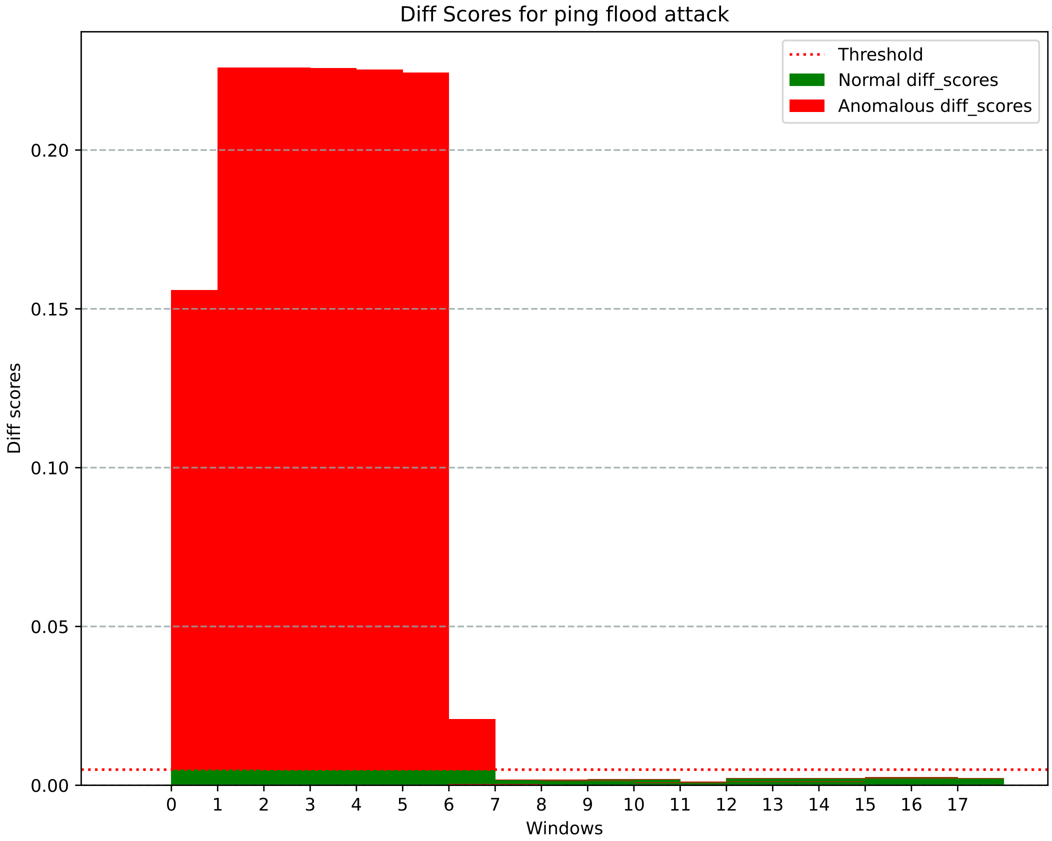
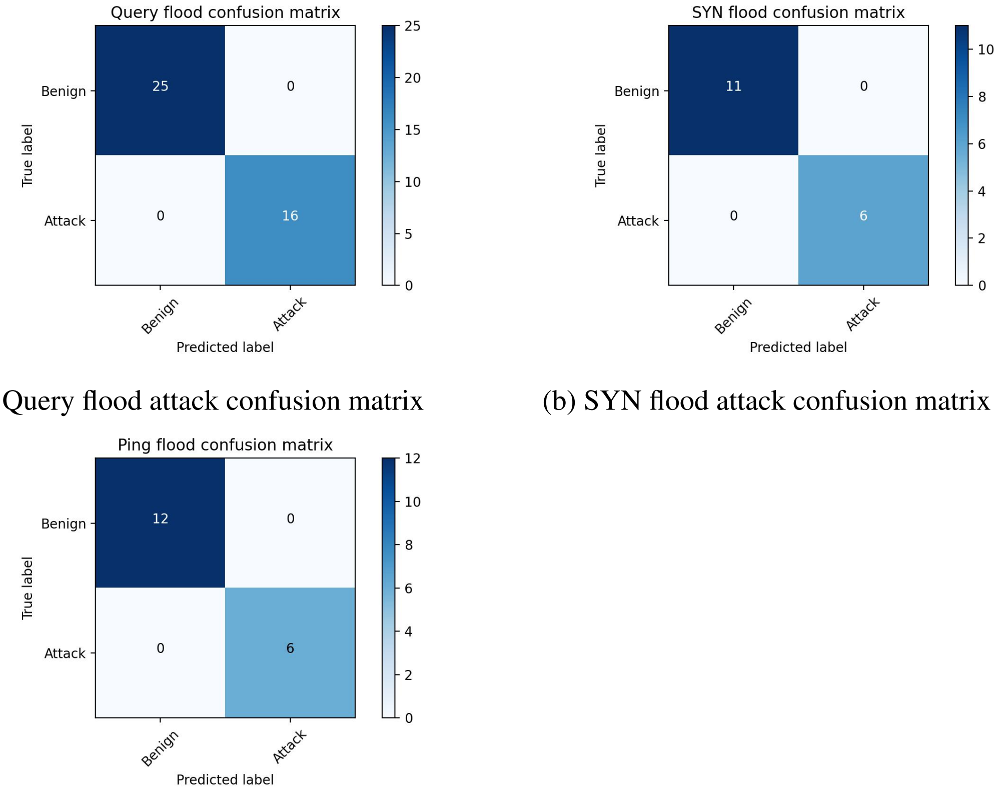


Figure 7.11: Diff Scores for Ping Flooding Attack Dataset

it leads to a high number of false alerts. Operators are often initially overwhelmed by the high number of false alerts, but with time, tend to start ignoring the alerts altogether – thereby defeating the purpose of the technology.

Results also proved that this method can be easily adapted to mitigate against concept drift in an industrial network. By simply changing the selected sliding window to a window representing the new normal traffic pattern, STADe can be used to continue detecting anomalies in the network without the need to train the model. This is a significant advantage over traditional anomaly detection techniques (i.e. machine learning). The advantages over machine learning anomaly detection methods are two-fold:

- There is no training requirement; and



(a) Query flood attack confusion matrix

(b) SYN flood attack confusion matrix

(c) Ping flood attack confusion matrix

Figure 7.12: Confusion Matrices for Query flood, SYN flood, and Ping flood attacks

- No requirement for large datasets.

These advantages are largely due to the low computational complexity of the method and its ease of deployment.

The remainder of this thesis outlines and reflects upon the research conclusions made across Chapters 2-7, and discusses potential future directions.

Chapter 8

Conclusions

Critical infrastructure and Operational Technology (OT) utilised in the oil and gas industry are becoming more exposed to cyber attacks due to technological advancements in Industrial Cyber-Physical Systems (ICPS) which is increasingly integrating OT systems with networking capabilities to communicate with the enterprise (IT) network. These technologies help sustain critical operations that affect our daily lives and usually operate by having sensors and actuators constantly communicating through an industrial network. However, there are many potential attacks that could target these ICPS (e.g. subsea control system) by compromising the industrial network as an initial step (survey in Chapter 2). To detect potential attacks on these industrial network, researchers have utilised misuse detection (investigated in Chapter 4) and Anomaly Detection (AD) techniques (investigated in Chapters 5 and 6). Nevertheless, misuse detection methods are unable to detect zero-day attacks (unknown attacks) while AD methods can, but with high false positive rates and in some cases, high computational overheads. This thesis focused on investigating and developing efficient attack detection methods on oil and gas industrial networks and proposed a novel method of anomaly detection. This chapter first summarises the findings of this thesis and then presents limitations and future work that could potentially extend the study.

8.1 Thesis summary

This thesis started by examining the growing threat of cyber-attacks to ICPS in the offshore oil and gas industry. The findings from this were highlighted in Chapter 2. The purpose of Chapter 2 was twofold, firstly, it examined the nature of oil and gas operations to aid our understanding of why it is a challenging sector to secure from cyber-attacks. Secondly, it provided insight into the history of cyber-attacks on oil and gas assets and existing detection techniques employed in previous research. As a result, it was shown that the oil and gas industry was more vulnerable to cyber-attacks when compared to other critical infrastructure industries (see Figure 2.2).

The oil and gas industry comprises three sub-sectors: upstream, downstream, and midstream. Previous studies showed that the upstream sector had been affected the most by cyber-attacks. Of these, the most frequent were 1) theft of operational information, and 2) Denial of Service (DoS) attacks as seen in Figure 2.5. These can be represented as attacks on Confidentiality and Availability respectively from the CIA triad (Confidentiality, Integrity, and Availability). However, in OT environments, Availability has a higher priority than Confidentiality or Integrity [75]. For this reason, this thesis focused mostly on detecting DoS attacks alongside attacks on integrity – where there was available data to use. To demonstrate the vulnerability of O&G systems to cyber-attacks, a case study of a typical subsea control system architecture was presented and analysed together with its vulnerabilities to DoS and spoofing attacks. Correlating these potential vulnerabilities to reported cyber attack incidents on upstream oil and gas assets revealed that attackers have the capabilities to exploit this system in its current state. Therefore, it has become important to develop effective methods to detect these kinds of attacks in such critical environments which, in turn, has led security researchers to carry out a number of studies on Intrusion Detection Systems (IDS). Many of these are machine learning (ML)-based Intrusion Detection Systems (IDSs) that can be broadly classified into supervised (signature-based or misuse detection) and unsupervised (anomaly detection) methods. Together with the current state of the art ML-based IDSs, due to the strong periodic patterns exhibited by industrial networks (discussed in Section 2.6.3), a further method based on packet timings was to be investigated as stated in the general methodology (Section 3).

Furthermore, Chapter 2 discussed the most widely used industrial protocols in the oil and gas industry and in Section 2.4.3 it was highlighted that ModbusTCP was by far the most deployed protocol. This laid the foundation for the experiments carried out throughout this research to be based on the ModbusTCP protocol.

One of the key findings in Chapter 2 was that despite the significant interest in detecting attacks in OT environments, not many studies have focused on the oil and gas industry due to a lack of data and testbeds with relevant hardware. This highlighted a need to investigate the effectiveness of supervised and unsupervised methods of detection in an oil and gas industrial environment and formed the motivation for the design and installation of a gas wellhead monitoring testbed presented in Chapter 4. Chapter 4 set out to investigate supervised ML-based detection on industrial networks. The questions Chapter 4 attempted to answer were how vulnerabilities in industrial communications protocols used in the OG industry (e.g. ModbusTCP protocol) could be exploited further and how well supervised ML methods would fare in detecting cyber-attacks in an oil and gas OT environment. To achieve this, a gas wellhead monitoring testbed was purposefully built, utilising a controller (RTU) specifically built for the use in offshore oil and gas platforms and the most popular industrial communication protocol used in the industry (ModbusTCP) [87, 88, 89] identified earlier in Chapter 2. Additionally, a novel field flooding attack was developed, which is capable of altering the structure of the modbus packet when injected into a network stream – leading to a DoS scenario. The impact of the field flooding attack was first evaluated on the gas wellhead monitoring test bed, then further evaluations were carried out on two additional testbeds representing different industry verticals. The purpose of testing on additional testbeds was to determine the impact of the novel attack on different industrial configurations. Figures 4.9, 4.10, and 4.11 showed that the hardware in the gas wellhead monitoring testbed was impacted the most by the attack. It caused a denial of service that lasted significantly longer (59 minutes) than what was obtained from testbeds 2 and 3 (see Table 4.5). As oil and gas operations involve handling volatile hydrocarbons in extreme conditions (high pressures and temperatures), a DoS lasting up to an hour could potentially have significant impact such as pipeline explosions leading to loss lives and damage to the environment.

Initial experiments to detect the field flooding attack involved the use of supervised ML algorithms. Eight (8) classifiers were evaluated and the best-performing classifier was XGBoost, which is an ensemble algorithm based on a decision tree meta-classifier. In fact, the top three performing classifiers were all tree-based algorithms (i.e. XGBoost, Random Forest, and Decision Trees) with F1 scores of 0.999, 0.998, and 0.98 respectively. These high F1 scores indicate that supervised ML-based IDSs are capable of detecting field flooding attacks on industrial networks. However, the limitation of this method is that supervised ML-based IDSs are unable to detect zero-day attacks and its success is also dependent on the availability of a properly labelled dataset to train the model – which is usually expensive and time-consuming to obtain.

To overcome the aforementioned limitations highlighted in Chapter 4, Chapter 5 sought to investigate unsupervised ML methods – which do not require labelled datasets and can detect zero-day attacks. However, one of the challenges of unsupervised ML methods is the generation of high false alerts. Therefore, the aim of Chapter 5 was to evaluate unsupervised ML algorithms and determine the method that generates the least false alerts which would form the foundation for further studies to improve the anomaly detection method. To achieve this, three algorithms from the most frequently used methods – distance-based (i.e. K-Nearest Neighbour, KNN), density-based (i.e. Local Outlier Factor, LOF), and model-based (i.e. isolation forest) methods – were used for the investigation and their performance was evaluated. The data used was collected from the gas wellhead monitoring testbed which contained field flooding, SYN flooding, and Man-in-the-Middle (MITM) attacks. The aim was to expand and vary the type of attacks to closely emulate the dynamic nature of a real attack. The algorithm that had the highest F1 score was isolation forest, obtaining a score of 0.673 while KNN and LOC scored 0.55 and 0.408 respectively.

Although Isolation Forest had the highest F1 score, it also had a high False Discovery Rate (FDR) of 39.33%. The implication of this is that within a 7-hour period (the period of data collection), the isolation forest model recorded 19,316 false alerts. This significant amount of false alerts would easily overwhelm security analysts and is a potential indication of why unsupervised anomaly detection IDSs are not popularly deployed in real environments. By contrast, the KNN algorithm, which is a distance-based method recorded only 5 false alerts (0.03% FDR) in the same period. It also had the highest precision score of 0.999. Although the recall score of 0.379 obtained by the KNN algorithm was low, it suggests that distance-based methods could potentially be improved upon to reduce incidents of false alerts – which is a significant hindrance to widespread adoption in the real world. The performance of unsupervised methods could also be potentially improved upon with the use of time windows. In this case, the model would be asked to classify a group of packets together to determine if the packet transmission rate and sequencing indicate an anomaly when compared to similar groups. This method would be capturing the behaviour of the system rather than analysing individual packets. Furthermore, the results in Chapter 5 showed that the problem of high false discovery rates in unsupervised ML anomaly detection could potentially be reduced using distance-based methods.

One key observation arising from the investigation of ML-based IDSs in Chapters 4 and 5 is that when the feature ranking filters were applied to the different datasets used, the feature `frame.time_delta` was ranked very high in all cases. This feature is a key parameter in characterising the periodicity of an industrial network as it represents the packet inter-arrival times. This was shown in Figures 4.12 and 5.1. The feature ranking filters are feature selection methods that score each feature based on its relevance to the attained label class. This served as an indication that the packet timings are highly relevant in detecting anomalies in industrial network communications as hypothesised in Section 2.6.3. Therefore, could the strong periodicity exhibited by industrial networks be exploited for use in anomaly detection methods? This is the question that Chapter 6 attempts to answer. In Chapter 6, a novel methodology of unsupervised Sliding Time-window Anomaly Detection (STADe) is developed and evaluated against the same datasets used in Chapter 5. The STADe approach and methodology (represented in Figure 6.1) involves, first, measuring the periodicity of the network, then second, detecting deviations from the periodicity.

The general idea is that the repetitive communication pattern of an industrial network could be likened to a heartbeat, and detecting breaks in repetition is similar to diagnosing an irregular heartbeat. Each industrial network communicates in its own rhythm (periodicity) and when measured effectively, deviations can be detected. This method depends on using a single feature – packet inter-arrival times δ – combined with standard deviation calculations to determine the periodicity in a selected time window. A distance-based metric, L2-norm (Euclidean distance) – similar to the KNN method – is then used to characterize the window behaviour in comparison to the sliding window. These approaches combined to form the core of the STADe methodology, are able to determine anomalous incidents in industrial network traffic. Additionally, the patterns in the segmented time windows can be visualised to give a human analyst context to the traffic behaviour.

The results of experiments carried out in Chapter 6 showed that the STADe methodology recorded F1 scores of 0.97, 0923, and 0.8 in detecting the field flooding, SYN flooding and MITM attacks respectively. More crucially, the model recorded FDR of 0 for all three attacks – meaning there were zero false positives. This constituted a significant improvement over other unsupervised ML detection approaches explored in Chapter 5, all of which were applied to the same dataset. It underscores the potential to measure the periodicity of industrial network communications and detect deviations from this periodic pattern in an unsupervised manner to develop an anomaly detection model capable of achieving high detection accuracy and low false alert rates. Such insights are pivotal for the development of an anomaly detection model, showcasing the capacity to achieve elevated levels of detection accuracy. Furthermore, the attack vectors in O&G subsystems including subsea control systems that were earlier identified in Chapter 2 were shown to be potentially detectable by STADe. Additional mitigation strategies in combination with other security measures were proposed to improve the security of subsea control systems and other related O&G subsystems. These proposed strategies would be in the form of an all inclusive defence-in-depth strategy to secure O&G systems.

In Chapter 7, a further evaluation of the STADe methodology was carried out to assess its consistency in achieving both low FDR and high detection scores in different industrial environments. To do this, a suitable public industrial dataset from a small-scale testbed emulating a process automation scenario containing three different DoS attacks was identified. Overall, the results of the experiments showed that every window containing attacks in the dataset was correctly identified with F1 scores of 1.0 and zero false discovery rates as shown in the confusion matrices in Figure 7.12.

To conclude, this thesis looked at the growing threats of cyber-attacks in the oil and gas industry and assessed supervised and unsupervised methods of detecting these attacks in an industrial network. A novel field flooding attack that has the capability of causing a DoS in an industrial network was developed alongside a supervised ML IDS that can detect such attacks with high accuracy. Furthermore, to overcome the limitations of supervised ML IDSs, unsupervised methods were investigated, leading to the development of a novel method, STADe, capable of detecting attacks with high detection and low false discovery rates. In doing these, this thesis made the following contributions:

C1 This thesis contributes an extensive survey on the cybersecurity challenges in the offshore oil and gas industry. The O&G production process and its vulnerabilities to cyber-attacks are described as well as the limitations in available datasets and testbeds for security research on OT infrastructure.

C2 Design and installation of a wellhead monitoring testbed to emulate the oil and gas production process and aid cybersecurity research in the OG industry.

C3 This research identifies a novel “Field Flooding” attack on the ModbusTCP protocol which can lead to a severe Denial of Service (DoS) attack.

C4 This research evaluates an Intrusion Detection System to effectively detect the Field Flooding attack on industrial control networks using a supervised machine learning approach.

C5 An investigation into how unsupervised machine learning algorithms can be utilised for anomaly detection in industrial cyber-physical systems.

C6 This research contributes a catalogue of labelled industrial network datasets in csv format including the original pcap files containing benign and attack data. The attacks carried out are field flooding attacks, SYN flooding attacks, and Man-in-the-Middle attacks.

C7 This research contributes a novel methodology of unsupervised Time-Series Method of Detecting Anomalies in Industrial Cyber-Physical Systems (ICPS) Networks.

C8 This research determines the adaptability of the STADe methodology described in Chapter 6 to different industrial cyber-physical networks.

This thesis has shown that the periodicity of industrial network communications can be measured and utilised in an unsupervised manner to achieve high detection and low false alert rates.

8.2 Limitations and Future directions

This thesis investigated and evaluated several supervised and unsupervised methods to detect various attacks. The scenarios explored were executed on a gas wellhead monitoring testbed purposefully built for the experiments. A key limitation of this work is the lack of access to larger-scaled testbeds containing multiple brands of PLCs, RTUs, and HMIs to closely emulate the diverse nature of a real offshore basic process control system. This would help in developing more robust detection models and could address the limitation in future research by creating larger datasets including multiple industrial communication protocols (e.g. OPC UA, EthernetIP, e.t.c.) and PLCs/RTUs. This would enhance the validity of the results in the sense that it would produce more generalisable results that are not only limited to the ModbusTCP protocol and would potentially be applicable to a broader scope of O&G operations. Also, it would serve as a basis to investigate if certain industrial protocols or attacks have a unique pattern that can be visually identifiable for security analysts. This would require the development of multiple systems integrated together – for example, a subsea Master Control Station connected to the gas wellhead control testbed – which could aid assessment of attack propagation through integrated systems and analysing its impact across multiple protocols.

One other key limitation specifically regarding the STADe methodology is that this approach would be more difficult to implement in an OT environment where there are intermittent control commands that may be part of the overall modus operandi of the plant. In other words, an OT environment that exhibits less periodicity than the norm. This may require a longer capture time to establish accurate baselines and thresholds for effective detection.

Overall, there are key insights from this research that could be adopted to potential use cases in the O&G industry. A significant one is that the STADe methodlogy could be deployed in subsea control communications networks to detect anomalies in the way the devices communicate. This could prove useful as these anomalies may not necessarily be as a result of a cyberattack, but rather as a result of the more common, but difficult to diagnose, problem of misconfigured communication devices on the industrial network. This would help not only security analysts, but engineers tasked with ensuring efficient communication of safety-critical devices. Furthermore, this research could also provide the basis for detecting changes in network communications latency. This would especially be useful in the ESD communications of a subsea control system as it is a safety-critical system that must maintain extremely low latency levels.

Another interesting potential for further research is investigating autonomic responses after attack detection. Such research may focus on the automatic identification of an attack pattern from which a model would be trained to trigger an optimal counter-response without the need for a human-in-the-loop. This is area is very broad and still in very early stages of development. Investigation of autonomic responses to cyber events in critical infrastructure can be further broken down as follows:

- Autonomic Response Strategies: It would be useful to investigate and develop advanced autonomic response strategies that can be integrated with tested anomaly detection systems in industrial environments. This could include adaptive incident response mechanisms that can automatically mitigate threats while minimising disruptions to operations.

- Resilience and Recovery: Investigate autonomic responses that not only focus on threat mitigation but also on system resilience and recovery. Developing strategies to autonomously restore critical services after an attack can reduce downtime and enhance overall system reliability.

- Resource Optimisation: Optimise resource allocation for autonomic responses. This could potentially be achieved by developing algorithms and strategies that prioritise responses based on the severity of threats, available resources, and potential impact on critical services.

- Adversarial Machine Learning: Research ways to make autonomic responses more robust against adversarial attacks. As attackers increasingly employ evasion techniques, developing countermeasures that can adapt to such tactics is essential. This aspect will keep increasing in priority as the autonomic response technology matures.

Bibliography

[1] DNV GL. *Oil and gas forecast to 2050*. 2017. URL: https://eto.dnvgl.com/2017/oilgas.

[2] Thumeera R Wanasinghe et al. “The Internet of Things in the Oil and Gas Industry: A Systematic Review”. In: *IEEE Internet of Things Journal* (2020).

[3] Giovanni Buizza Avanzini, Andrea Spessa, et al. “Cybersecurity Verification Approach for the Oil & Gas Industry”. In: *Offshore Mediterranean Conference and Exhibition*. Offshore Mediterranean Conference. 2019.

[4] Dorothy Bundi and Mayieka Jared Maranga. “EFFECTS OF CYBERCRIME ON OIL AND GAS INDUSTRY”. In: *GSJ* 8.6 (2020).

[5] Zakarya Drias, Ahmed Serhrouchni, and Olivier Vogel. “Analysis of cyber security for industrial control systems”. In: *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE. 2015, pp. 1–8.

[6] George Stergiopoulos, Dimitris A Gritzalis, and Evangelos Limnaios. “Cyber-Attacks on the Oil & Gas Sector: A Survey on Incident Assessment and Attack Patterns”. In: *IEEE Access* 8 (2020), pp. 128440–128475.

[7] Hongfang Lu et al. “Oil and Gas 4.0 era: A systematic review and outlook”. In: *Computers in Industry* 111 (2019), pp. 68–90.

[8] Iosif Progoulakis et al. “Perspectives on Cyber Security for Offshore Oil and Gas Assets”. In: *Journal of Marine Science and Engineering* 9.2 (2021), p. 112.

[9] Keith Stouffer, Joe Falco, and Karen Scarfone. "Guide to industrial control systems (ICS) security". In: *NIST special publication* 800.82 (2011), pp. 16–16.

[10] Karstein Berge Kristiansen. "Digitalization goes subsea". In: *Offshore Technology Conference*. OnePetro. 2019.

[11] Karstein Berge Kristiansen and Sigbjørn Mæland. "Benefits of Removing Subsea Control Boundaries". In: *Offshore Technology Conference*. OnePetro. 2020.

[12] Bruce Bailie and Matthew Chinn. "Effectively harnessing data to navigate the new normal: Overcoming the barriers of digital adoption". In: *Offshore Technology Conference*. OnePetro. 2018.

[13] Tarek Gaber et al. "Autonomous haulage systems in the mining industry: Cybersecurity, communication and safety issues and challenges". In: *Electronics* 10.11 (2021), p. 1357.

[14] Dong-Seong Kim et al. "An overview on industrial control networks". In: *Industrial Sensors and Controls in Communication Networks: From Wired Technologies to Cloud Computing and the Internet of Things* (2019), pp. 3–16.

[15] Peter Huitsing et al. "Attack taxonomies for the Modbus protocols". In: *International Journal of Critical Infrastructure Protection* 1 (2008), pp. 37–44.

[16] Xinchi He et al. "Anomaly detection sensors for a modbus-based oil and gas well-monitoring system". In: *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*. IEEE. 2019, pp. 1–8.

[17] Leandros Maglaras. "Intrusion detection in scada systems using machine learning techniques". PhD thesis. University of Huddersfield, 2018.

[18] Hervé Debar, Marc Dacier, and Andreas Wespi. "Towards a taxonomy of intrusion-detection systems". In: *Computer networks* 31.8 (1999), pp. 805–822.

[19] Xavier Larriva-Novo et al. "Efficient distributed preprocessing model for machine learning-based anomaly detection over large-scale cybersecurity datasets". In: *Applied Sciences* 10.10 (2020), p. 3430.

[20] Harsh Chaudhary et al. “A review of various challenges in cybersecurity using artificial intelligence”. In: *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE. 2020, pp. 829–836.

[21] Rahul Kale and Vrizlynn LL Thing. “Few-shot weakly-supervised cybersecurity anomaly detection”. In: *Computers & Security* 130 (2023), p. 103194.

[22] ANM Bazlur Rashid et al. “Anomaly detection in cybersecurity datasets via cooperative co-evolution-based feature selection”. In: *ACM Transactions on Management Information Systems (TMIS)* 13.3 (2022), pp. 1–39.

[23] Luis Rosa et al. “Intrusion and anomaly detection for the next-generation of industrial automation and control systems”. In: *Future Generation Computer Systems* 119 (2021), pp. 50–67.

[24] Tommaso Zoppi, Andrea Ceccarelli, and Andrea Bondavalli. “Unsupervised algorithms to detect zero-day attacks: Strategy and application”. In: *Ieee Access* 9 (2021), pp. 90603–90615.

[25] Hakan Kayan et al. “Cybersecurity of Industrial Cyber-Physical Systems: A Review”. In: *arXiv preprint arXiv:2101.03564* (2021).

[26] Stig Olav Settemsdal, Ben Bishop, et al. “When to go with cloud or edge computing in offshore oil and gas”. In: *SPE Offshore Europe Conference and Exhibition*. Society of Petroleum Engineers. 2019.

[27] Cristina Alcaraz and Sherali Zeadally. “Critical control system protection in the 21st century”. In: *Computer* 46.10 (2013), pp. 74–83.

[28] Ioannis Stellios et al. “A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services”. In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 3453–3495.

[29] M Nygaard and S Mukhopadyay. *Dragonstone Strategy Kickoff Report*. Tech. rep. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.

[30] Hansong Xu et al. “A survey on industrial Internet of Things: A cyber-physical systems perspective”. In: *IEEE Access* 6 (2018), pp. 78238–78259.

[31] Planete Energies. *Offshore Oil and Gas Production*. 2015. URL: https://www.planete-energies.com/en/medias/close/offshore-oil-and-gas-production.

[32] Ruth Olurounbi. *Nigeria in talks with Shell over onshore divestment plans*. 2021. URL: https://www.bloomberg.com/news/articles/2021-05-18/nigeria-in-talks-with-shell-over-onshore-divestment-plans.

[33] Hellenic Shipping News. *Nigeria confirms it is in talks with Shell over sale of onshore oil assets*. 2021. URL: https://www.hellenicshippingnews.com/nigeria-confirms-it-is-in-talks-with-shell-over-sale-of-onshore-oil-assets/.

[34] Equinor. *Equinor sells its US onshore assets in the Bakken*. 2021. URL: https://www.equinor.com/en/news/20210210-us-onshore.html.

[35] Piotr Ciepiela. *Digitization and Cyber Disruption in Oil and Gas*. 2016. URL: https://www.ey.com/Publication/vwLUAssets/ey-wpc-digitization-and-cyber/FILE/ey-wpc-digitization-and-cyber.pdf.

[36] Ben Dickinson, Mario Chiock, et al. "Guest Editorial: Countering Security Issues in the Digital World". In: *Journal of Petroleum Technology* 71.06 (2019), pp. 14–15.

[37] Space Rogue. *Tilting It Sideways*. 2016. URL: https://www.spacerogue.net/wordpress/?p=625.

[38] Robert Lee, Michael Assante, and Tim Conway. *Media report of the Baku-Tbilisi-Ceyhan (BTC) pipeline Cyber Attack*. 2014. URL: https://ics.sans.org/media/Media-report-of-the-BTC-pipeline-Cyber-Attack.pdf.

[39] Jordan Robertson and Michael Riley. *Mysterious '08 Turkey Pipeline Blast Opened New Cyberwar*. 2014. URL: https://www.bloomberg.com/news/articles/2014-12-10/mysterious-08-turkey-pipeline-blast-opened-new-cyberwar.

[40] Collin Eaton. *HACKED Part 1: As cyberattacks become more sophisticated, energy industry's controls provide an alluring target*. 2018. URL: https://www.houstonchronicle.com/news/houston-texas/houston/article/As-cyberattacks-become-more-sophisticated-energy-10973429.php.

[41] Ponemon Institute. *The State of Cybersecurity in the Oil & Gas Industry: United States*. 2017. URL: https://assets.new.siemens.com/siemens/assets/api/uuid:949a55b50ef511c8f0d32621c6a428bb2d3d95ee/version:1516999356/ps-cd-ponemon-study-state-of-cybersecurity-oil-and-gas-en.pdf.

[42] David Sanger, Clifford Krauss, and Nicole Perlroth. *Cyberattack forces a shutdown of a top US pipeline*. 2021. URL: https://www.nytimes.com/2021/05/08/us/politics/cyberattack-colonial-pipeline.html.

[43] Joe R Reeder and Cadet Tommy Hall. “Cybersecurity’s Pearl Harbor Moment: Lessons Learned from the Colonial Pipeline Ransomware Attack”. In: (2021).

[44] Kimberly Wood. *Cybersecurity Policy Responses to the Colonial Pipeline Ransomware Attack*. 2023. URL: https://www.law.georgetown.edu/environmental-law-review/blog/cybersecurity-policy-responses-to-the-colonial-pipeline-ransomware-attack/.

[45] Kyoung-Dae Kim and Panganamala R Kumar. “Cyber–physical systems: A perspective at the centennial”. In: *Proceedings of the IEEE* 100.Special Centennial Issue (2012), pp. 1287–1308.

[46] Maryna Krotofil and Dieter Gollmann. “Industrial control systems security: What is happening?” In: *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE. 2013, pp. 670–675.

[47] Yilin Mo et al. “Cyber–physical security of a smart grid infrastructure”. In: *Proceedings of the IEEE* 100.1 (2011), pp. 195–209.

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

[49] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. “Security and privacy challenges in industrial internet of things”. In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE. 2015, pp. 1–6.

[50] Wazir Zada Khan et al. “A reliable Internet of Things based architecture for oil and gas industry”. In: *2017 19th International conference on advanced communication Technology (ICACT)*. IEEE. 2017, pp. 705–710.

[51] Naoum Sayegh et al. “Internal security attacks on SCADA systems”. In: *2013 Third International Conference on Communications and Information Technology (ICCIT)*. IEEE. 2013, pp. 22–27.

[52] Sajid Nazir, Shushma Patel, and Dilip Patel. “Assessing and augmenting SCADA cyber security: A survey of techniques”. In: *Computers & Security* 70 (2017), pp. 436–454.

[53] Deval Bhamare et al. “Cybersecurity for industrial control systems: A survey”. In: *Computers & Security* 89 (2020), p. 101677.

[54] Bill Miller and Dale Rowe. “A survey SCADA of and critical infrastructure incidents”. In: *Proceedings of the 1st Annual conference on Research in information technology*. 2012, pp. 51–56.

[55] Jairo Giraldo et al. “Security and privacy in cyber-physical systems: A survey of surveys”. In: *IEEE Design & Test* 34.4 (2017), pp. 7–17.

[56] Sean McBride, Jeffery Ashcraft, and Nathan Belk. *Overload: Critical Lessons from 15 Years of ICS Vulnerabilities*. 2016. URL: https://www.fireeye.com/blog/threat-research/2016/08/overload-critical-lessons-from-15-years-of-ics-vulnerabilities.html.

[57] Dragos Inc. *Global Oil and Gas Cyber Threat Perspective*. 2019. URL: https://www.dragos.com/wp-content/uploads/Dragos-Oil-and-Gas-Threat-Perspective-2019.pdf.

[58] Francis Lobo. *Upstream Oil & Gas Cyber Risk: Insurance Technical Review*. 2019. URL: https://www.lmalloyds.com/LMA/publications/upstreamcyberreport.aspx.

[59] Anil Lamba. “Protecting ‘Cybersecurity & Resiliency’of Nation’s Critical Infrastructure–Energy, Oil & Gas”. In: *International Journal of Current Research* 10 (2018), pp. 76865–76876.

[60] Emily Darko. “Short guide summarising the oil and gas industry lifecycle for a non-technical audience”. In: *London: Overseas Development Institute* (2014).

[61] Ann Scarborough Bull and Milton S Love. “Worldwide oil and gas platform decommissioning: a review of practices and reefing options”. In: *Ocean & coastal management* 168 (2019), pp. 274–306.

[62] Alexander Polyakov and Matheu Geli. *SAP Cybersecurity for Oil and Gas*. Tech. rep. ERP Scan, 2015.

[63] Håvard Devold. “OIL AND GAS PRODUCTION HANDBOOK”. In: (2006).

[64] A Mittal, A Slaughter, and P Zonneveld. “Protecting the connected barrels: Cybersecurity for upstream oil and gas”. In: *Deloitte Insights, London, UK, Tech. Rep* (2017).

[65] Xiaodao Chen et al. “Offshore oil spill monitoring and detection: Improving risk management for offshore petroleum cyber-physical systems”. In: *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE. 2017, pp. 841–846.

[66] Jaime HuiChoo Tan et al. “Transforming Offshore Oil and Gas Production Platforms into Smart Unmanned Installations”. In: *Offshore Technology Conference Asia*. OnePetro. 2020.

[67] Utheswaran Krishna Moorthy et al. “Alternative Method to Supply Pneumatic Air to an Unmanned Platform, In the Event of the Platform’s Instrument Gas System is on Downtime”. In: *SPE Annual Technical Conference and Exhibition*. OnePetro. 2020.

[68] MF Mahmoud Radwan. “Safe and Economic Attractive Rigless Operations Using a Digital Slickline in Unmanned Platform with Low Structure Loads and Spacing”. In: *Abu Dhabi International Petroleum Exhibition & Conference*. OnePetro. 2020.

[69] Justin Okpala et al. “Enabling Reservoir Management Excellence through Real Time Surveillance of an Unmanned Onshore Gas-Condensate Field Platform”. In: *SPE Nigeria Annual International Conference and Exhibition*. OnePetro. 2020.

[70] Jan Erik Vinnem. “Assessment of risk tolerance for future autonomous offshore installations”. In: *Safety science* 134 (2021), p. 105059.

[71] Anna Isabella Thomassen Frostad et al. “Unmanned Full Processing Platforms; Using Subsea Technology as Enabler”. In: *Offshore Technology Conference*. Offshore Technology Conference. 2020.

[72] Oil & Gas Authority. *Analysis of UKCS Operating Costs in 2016*. 2017. URL: https://www.ogauthority.co.uk/media/4514/ukcs-operating-cost-analysis.pdf.

[73] Fan Zhang et al. “Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data”. In: *IEEE Transactions on Industrial Informatics* 15.7 (2019), pp. 4362–4369.

[74] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. “A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT”. In: *Journal of Network and Computer Applications* 149 (2020), p. 102481.

[75] AA Baybulatov and VG Promyslov. “Cybersecurity assessment using delay from backlog bound calculation”. In: *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE. 2020, pp. 1–6.

[76] Christian Mudrak. “Subsea production systems-A review of components, maintenance and reliability”. PhD thesis. University of Leoben, 2016.

[77] Eirini Anthi et al. “Secure data sharing and analysis in cloud-based energy management systems”. In: *Cloud Infrastructures, Services, and IoT Systems for Smart Cities*. Springer, 2017, pp. 228–242.

[78] Namje Park and Namhi Kang. “Mutual authentication scheme in secure internet of things technology for comfortable lifestyle”. In: *Sensors* 16.1 (2016), p. 20.

[79] Joe Weiss. *What the lack of cyber security of process sensors means*. 2019. URL: https://sigasec.com/blog/uncategorized/what-the-lack-of-cyber-security-of-process-sensors-means/.

[80] Helen K White et al. “Impact of the Deepwater Horizon oil spill on a deepwater coral community in the Gulf of Mexico”. In: *Proceedings of the National Academy of Sciences* 109.50 (2012), pp. 20303–20308.

[81] Jairo Giraldo et al. “A survey of physics-based attack detection in cyber-physical systems”. In: *ACM Computing Surveys (CSUR)* 51.4 (2018), pp. 1–36.

[82] Mazen Azzam et al. “Grounds for Suspicion: Physics-based Early Warnings for Stealthy Attacks on Industrial Control Systems”. In: *arXiv preprint arXiv:2106.07980* (2021).

[83] James M Taylor and Hamid R Sharif. “Security challenges and methods for protecting critical infrastructure cyber-physical systems”. In: *2017 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*. IEEE. 2017, pp. 1–6.

[84] Sabrina Sicari et al. “REATO: REActing TO Denial of Service attacks in the Internet of Things”. In: *Computer Networks* 137 (2018), pp. 37–48.

[85] Mordor Intelligence. *Subsea Systems Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026)*. 2021. URL: https://www.mordorintelligence.com/industry-reports/subsea-systems-market.

[86] Research and Market. *Global Distributed Control Systems (DCS) Market in the Oil and Gas Industry 2019-2023*. Tech. rep. Research and Market, 2019.

[87] Marcio Andrey Teixeira et al. “SCADA system testbed for cybersecurity research using machine learning approach”. In: *Future Internet* 10.8 (2018), p. 76.

[88] Mohammad E Alim, Shelton Wright, and Tommy Morris. “A Laboratory-Scale Spillway SCADA System Testbed for Cybersecurity Research”. In: ().

[89] Igor Nai Fovino et al. “An experimental investigation of malware attacks on SCADA systems”. In: *International Journal of Critical Infrastructure Protection* 2.4 (2009), pp. 139–145.

[90] F Hacquebord and C Pernet. “Drilling Deep: A look at Cyberattacks on the Oil and Gas Industry”. In: *Trend Micro Research* (2019).

[91] David Kravets. *Feds: Hacker Disabled Offshore Oil Platforms’ Leak-Detection System*. 2009. URL: https://www.wired.com/2009/03/feds-hacker-dis/.

[92] Piotr Ciepiela, Bala V Venkateshwaran, et al. "Evolution of Cyber Threats and the Development of New Security Architecture". In: *22nd World Petroleum Congress*. World Petroleum Congress. 2017.

[93] Science X. *Chevron says hit by Stuxnet virus in 2010*. 2012. URL: https://phys.org/news/2012-11-chevron-stuxnet-virus.html.

[94] Rahat Masood. "Assessment of cyber security challenges in nuclear power plants security incidents, threats, and initiatives". In: *Cybersecurity and Privacy Research Institute the George Washington University* (2016).

[95] Bernard Brode. *7 cyber threat actors to watch for in 2021*. 2021. URL: https://www.securityinfowatch.com/cybersecurity/article/21207268/7-cyber-threat-actors-to-watch-for-in-2021.

[96] A Stacey, M Birkinshaw, and JV Sharp. "Life extension issues for ageing offshore installations". In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 48227. 2008, pp. 199–215.

[97] M Rosner, P Herve, K Moore, et al. "Using a Cognitive Analytic Approach to Enhance Cybersecurity on Oil and Gas OT Systems". In: *Offshore Technology Conference*. Offshore Technology Conference. 2017.

[98] Hadi Almusaher, Gulzar Alam, et al. "How Feasible Moving Target Defense is Within ICS Environment". In: *International Petroleum Technology Conference*. International Petroleum Technology Conference. 2020.

[99] Rachel Adams-Heard, David Wethe, and Kevin Crowley. *Turning Oil Wells Back on Is Trickier Than Shutting Them Off*. 2020. URL: https://energynow.com/2020/05/turning-oil-wells-back-on-is-trickier-than-shutting-them-off/.

[100] Doug Walser. *Production Restarts: Fiscal, Technical Issues Define Operator Strategies In Restarting Shut-In Wells*. 2021. URL: https://www.aogr.com/magazine/cover-story/fiscal-technical-issues-define-operator-strategies-in-restarting-shut-in-wells.

[101] Offshore Technology. *The longest standing fixed offshore platforms*. 2019. URL: https://www.offshore-technology.com/features/the-longest-standing-fixed-offshore-platforms/.

[102] Silvia Tham. *Exploring the growth of the FPSO industry*. 2019. URL: https://www.fircroft.com/blogs/exploring-the-growth-of-the-fpso-industry-95120129599.

[103] Oil and Gas IQ. *10 reasons why FPSOs are the future of oil and gas*. 2019. URL: https://www.oilandgasiq.com/oil-gas/news/ten-reasons-why-fpsos-are-the-future-of-oil-and-ga.

[104] Offshore Energy. *Samsung to Build Largest FPSO in the World*. 2013. URL: https://www.offshore-energy.biz/samsung-to-build-largest-fpso-in-the-world/.

[105] Karsten Friis, Lilly Pijnenburg Muller, and Lars Gjesvik. "Cyber-weapons in International Politics: Possible sabotage against the Norwegian petroleum sector". In: *NUPI Report* (2018).

[106] Alireza Esfahani et al. "A lightweight authentication mechanism for M2M communications in industrial IoT environment". In: *IEEE Internet of Things Journal* 6.1 (2017), pp. 288–296.

[107] Jangirala Srinivas et al. "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things". In: *IEEE Transactions on Dependable and Secure Computing* 17.6 (2018), pp. 1133–1146.

[108] Liangchen Chen, Shu Gao, and Baoxu Liu. "An improved density peaks clustering algorithm based on grid screening and mutual neighborhood degree for network anomaly detection". In: *Scientific Reports* 12.1 (2022), p. 1409.

[109] Shamshair Ali et al. "Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection". In: *Electronics* 11.23 (2022), p. 3934.

[110] Shilin He et al. "Experience report: System log analysis for anomaly detection". In: *2016 IEEE 27th international symposium on software reliability engineering (ISSRE)*. IEEE. 2016, pp. 207–218.

[111] Majjed Al-Qatf et al. “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection”. In: *Ieee Access* 6 (2018), pp. 52843–52856.

[112] Hanan Hindy et al. “A taxonomy of network threats and the effect of current datasets on intrusion detection systems”. In: *IEEE Access* 8 (2020), pp. 104650–104675.

[113] Khloud Al Jallad, Mohamad Aljnidi, and Mohammad Said Desouki. “Anomaly detection optimization using big data and deep learning to reduce false-positive”. In: *Journal of Big Data* 7.1 (2020), pp. 1–12.

[114] Michael Zipperle et al. “Provenance-based intrusion detection systems: A survey”. In: *ACM Computing Surveys* 55.7 (2022), pp. 1–36.

[115] Stefan Axelsson. “The base-rate fallacy and the difficulty of intrusion detection”. In: *ACM Transactions on Information and System Security (TISSEC)* 3.3 (2000), pp. 186–205.

[116] J Sadowski. *Zero Tolerance: More Zero-Days Exploited in 2021 Than Ever Before*. 2022.

[117] Felix O Olowononi, Danda B Rawat, and Chunmei Liu. “Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS”. In: *IEEE Communications Surveys & Tutorials* (2020).

[118] Mohammad Sayad Haghighi, Faezeh Farivar, and Alireza Jolfaei. “A machine learning-based approach to build zero false-positive ipss for industrial iot and cps with a case study on power grids security”. In: *IEEE Transactions on Industry Applications* (2020).

[119] Eirini Anthi et al. “Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks”. In: *computers & security* 108 (2021), p. 102352.

[120] Pu Zeng and Peng Zhou. “Intrusion Detection in SCADA System: A Survey”. In: *Intelligent Computing and Internet of Things*. Springer, 2018, pp. 342–351.

[121] Louis Wehenkel. “Machine learning approaches to power-system security assessment”. In: *IEEE Expert* 12.5 (1997), pp. 60–72.

[122] Sumeet Dua and Xian Du. *Data mining and machine learning in cybersecurity*. CRC press, 2016.

[123] Alvaro A Cárdenas et al. “Attacks against process control systems: risk assessment, detection, and response”. In: *Proceedings of the 6th ACM symposium on information, computer and communications security*. 2011, pp. 355–366.

[124] Yun-Gui Zhang et al. “SCADA intrusion detection system based on self-learning Semi-Supervised One-Class Support Vector Machine”. In: *Metallurgical Industry Automation* 37.2 (2013), pp. 1–5.

[125] SLP Yasakethu and J Jiang. “Intrusion detection via machine learning for SCADA system protection”. In: *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*. 2013, pp. 101–105.

[126] Justin M Beaver, Raymond C Borges-Hink, and Mark A Buckner. “An evaluation of machine learning methods to detect malicious SCADA communications”. In: *2013 12th International Conference on Machine Learning and Applications*. Vol. 2. IEEE. 2013, pp. 54–59.

[127] Leandros A Maglaras and Jianmin Jiang. “Intrusion detection in SCADA systems using machine learning techniques”. In: *2014 Science and Information Conference*. IEEE. 2014, pp. 626–631.

[128] Raymond C Borges Hink et al. “Machine learning for power system disturbance and cyber-attack discrimination”. In: *2014 7th International symposium on resilient control systems (ISRCS)*. IEEE. 2014, pp. 1–8.

[129] Noam Erez and Avishai Wool. “Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems”. In: *International Journal of Critical Infrastructure Protection* 10 (2015), pp. 59–70.

[130] Vojtech Franc, Michal Sofka, and Karel Bartos. “Learning detector of malicious network traffic from weak labels”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2015, pp. 85–99.

[131] Patric Nader, Paul Honeine, and Pierre Beauseroy. “Detection of cyberattacks in a water distribution system using machine learning techniques”. In: *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*. IEEE. 2016, pp. 25–30.

[132] Kevin Leahy et al. “Diagnosing wind turbine faults using machine learning techniques applied to operational data”. In: *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2016, pp. 1–8.

[133] Alfonso Valdes, Richard Macwan, and Matthew Backes. “Anomaly detection in electrical substation circuits via unsupervised machine learning”. In: *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*. IEEE. 2016, pp. 500–505.

[134] Kyriakos Stefanidis and Artemios G Voyiatzis. “An HMM-based anomaly detection approach for SCADA systems”. In: *IFIP International Conference on Information Security Theory and Practice*. Springer. 2016, pp. 85–99.

[135] Karel Bartos, Michal Sofka, and Vojtech Franc. “Optimized invariant representation of network traffic for detecting unseen malware variants”. In: *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 2016, pp. 807–822.

[136] Youbiao He, Gihan J Mendis, and Jin Wei. “Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism”. In: *IEEE Transactions on Smart Grid* 8.5 (2017), pp. 2505–2516.

[137] Sasanka Potluri, Navin Francis Henry, and Christian Diedrich. “Evaluation of hybrid deep learning techniques for ensuring security in networked control systems”. In: *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE. 2017, pp. 1–8.

[138] Anastasis Keliris et al. “Machine learning-based defense against process-aware attacks on industrial control systems”. In: *2016 IEEE International Test Conference (ITC)*. IEEE. 2016, pp. 1–10.

[139] Yi Zhang, Marija D Ilić, and Ozan K Tonguz. “Mitigating blackouts via smart relays: A machine learning approach”. In: *Proceedings of the IEEE* 99.1 (2010), pp. 94–118.

[140] Imtiaz Ullah and Qusay H Mahmoud. “A hybrid model for anomaly-based intrusion detection in SCADA networks”. In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 2160–2167.

[141] Irfan A Siddavatam et al. “An ensemble learning for anomaly identification in SCADA system”. In: *2017 7th International Conference on Power Systems (ICPS)*. IEEE. 2017, pp. 457–462.

[142] Thiago Alves, Rishabh Das, and Thomas Morris. “Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers”. In: *IEEE Embedded Systems Letters* 10.3 (2018), pp. 99–102.

[143] Oliver Eigner, Philipp Kreimel, and Paul Tavolato. “Detection of man-in-the-middle attacks on industrial control networks”. In: *2016 International Conference on Software Security and Assurance (ICSSA)*. IEEE. 2016, pp. 64–69.

[144] Maede Zolanvari et al. “Machine learning-based network vulnerability analysis of industrial Internet of Things”. In: *IEEE Internet of Things Journal* 6.4 (2019), pp. 6822–6834.

[145] Seungoh Choi, Jeong-Han Yun, and Sin-Kyu Kim. “A comparison of ICS datasets for security research based on attack paths”. In: *International Conference on Critical Information Infrastructures Security*. Springer. 2018, pp. 154–166.

[146] Thomas Morris and Wei Gao. “Industrial control system traffic data sets for intrusion detection research”. In: *International Conference on Critical Infrastructure Protection*. Springer. 2014, pp. 65–78.

[147] Antoine Lemay and José M Fernandez. “Providing {SCADA} network data sets for intrusion detection research”. In: *9th Workshop on Cyber Security Experimentation and Test ({CSET} 16)*. 2016.

[148] Jonathan Goh et al. “A dataset to support research in the design of secure water treatment systems”. In: *International conference on critical information infrastructures security*. Springer. 2016, pp. 88–99.

[149] Nicholas R Rodofile et al. “Process control cyber-attacks and labelled datasets on S7Comm critical infrastructure”. In: *Australasian Conference on Information Security and Privacy*. Springer. 2017, pp. 452–459.

[150] ICS Lab. *4SICS ICS Lab PCAP Files*. 2015. URL: https://www.netresec.com/?page=PCAP4SICS.

[151] Dale Peterson and Reid Wightman. *Digital bond S4x15 ICS village CTF PCAP files*. 2015. URL: https://www.netresec.com/?page=DigitalBond_S4.

[152] DEFCON23. *compilation of ICS PCAP files indexed by protocol*. 2015. URL: https://media.defcon.org/DEFCON23/DEFCON23villages/DEFCON23icsvillage/DEFCON23ICSVillagepacketcaptures.rar.

[153] Kevser Ovaz Akpinar and Ibrahim Ozcelik. “Methodology to Determine the Device-Level Periodicity for Anomaly Detection in EtherCAT-Based Industrial Control Network”. In: *IEEE Transactions on Network and Service Management* 18.2 (2020), pp. 2308–2319.

[154] Rafael Ramos Regis Barbosa, Ramin Sadre, and Aiko Pras. “Exploiting traffic periodicity in industrial control networks”. In: *International journal of critical infrastructure protection* 13 (2016), pp. 52–62.

[155] Michael Bailey et al. “The menlo report”. In: *IEEE Security & Privacy* 10.2 (2012), pp. 71–75.

[156] Abubakar Sadiq Mohammed et al. “Cybersecurity Challenges in the Offshore Oil and Gas Industry: An Industrial Cyber-Physical Systems (ICPS) Perspective”. In: *ACM Trans. Cyber-Phys. Syst.* (2022). URL: https://doi.org/10.1145/3548691.

[157] Jesus Gonzalez and Mauricio Papa. “Passive scanning in Modbus networks”. In: *International Conference on Critical Infrastructure Protection*. Springer. 2007, pp. 175–187.

[158] RTU RS. *485 Specification, Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev*. 2002.

[159] Haseeb Ahmed Chattha et al. “Implementation of Cyber-Physical Systems with Modbus Communication for Security Studies”. In: *2021 International Conference on Cyber Warfare and Security (ICCWS)*. IEEE. 2021, pp. 45–50.

[160] John Luswata et al. “Analysis of scada security using penetration testing: A case study on modbus tcp protocol”. In: *2018 29th Biennial Symposium on Communications (BSC)*. IEEE. 2018, pp. 1–5.

[161] Christopher Parian, Terry Guldimann, and Sajal Bhatia. “Fooling the master: Exploiting weaknesses in the Modbus protocol”. In: *Procedia Computer Science* 171 (2020), pp. 2453–2458.

[162] Penke Satyanarayana et al. “Detection and Blocking of Replay, False Command, and False Access Injection Commands in SCADA Systems with Modbus Protocol”. In: *Security and Communication Networks* 2021 (2021).

[163] May Bashendy et al. “Design and implementation of cyber-physical attacks on modbus/tcp protocol”. In: *World Congress on Industrial Control Systems Security (WCICSS-2020)*. 2020.

[164] John Stranahan, Tapan Soni, and Vahid Heydari. “Supervisory control and data acquisition testbed vulnerabilities and attacks”. In: *2019 SoutheastCon*. IEEE. 2019, pp. 1–5.

[165] Cristina Alcaraz et al. “Covert channels-based stealth attacks in industry 4.0”. In: *IEEE Systems Journal* 13.4 (2019), pp. 3980–3988.

[166] Panagiotis Radoglou Grammatikis et al. “ARIES: a novel multivariate intrusion detection system for smart grid”. In: *Sensors* 20.18 (2020), p. 5305.

[167] Mahdis Saharkhizan et al. “An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic”. In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 8852–8859.

[168] Thomas H Morris et al. “Deterministic intrusion detection rules for MODBUS protocols”. In: *2013 46th Hawaii International Conference on System Sciences*. IEEE. 2013, pp. 1773–1781.

[169] F Katulić et al. “Enhancing Modbus/TCP-Based Industrial Automation and Control Systems Cybersecurity Using a Misuse-Based Intrusion Detection System”. In: *2022 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*. IEEE. 2022, pp. 964–969.

[170] Michael J Assante and Robert M Lee. “The industrial control system cyber kill chain”. In: *SANS Institute InfoSec Reading Room* 1 (2015).

[171] Panagiotis Radoglou-Grammatikis et al. “Implementation and detection of modbus cyberattacks”. In: *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE. 2020, pp. 1–4.

[172] R Rohith, Minal Moharir, G Shobha, et al. “SCAPY-A powerful interactive packet manipulation program”. In: *2018 international conference on networking, embedded and wireless systems (ICNEWS)*. IEEE. 2018, pp. 1–5.

[173] Gerald Combs. *Wireshark*. 2022. URL: https://www.wireshark.org/.

[174] AdvancedHMI. *HMI software by AdvancedHMI, the industry’s most flexible HMI*. 2022. URL: https://www.advancedhmi.com/.

[175] Otis Alexander, Misha Belisle, and Jacob Steele. “MITRE ATT&CK® for industrial control systems: Design and philosophy”. In: *The MITRE Corporation: Bedford, MA, USA* (2020).

[176] MohammadNoor Injadat et al. “Bayesian optimization with machine learning algorithms towards anomaly detection”. In: *2018 IEEE global communications conference (GLOBECOM)*. IEEE. 2018, pp. 1–6.

[177] Eirini Anthi et al. “A three-tiered intrusion detection system for industrial control systems”. In: *Journal of Cybersecurity* 7.1 (2021), tyab006.

[178] Mark Hall et al. “The WEKA data mining software: an update”. In: *ACM SIGKDD explorations newsletter* 11.1 (2009), pp. 10–18.

[179] Ahmed Mahfouz et al. “Ensemble classifiers for network intrusion detection using a novel network attack dataset”. In: *Future Internet* 12.11 (2020), p. 180.

[180] Joffrey L Leevy et al. “Detecting information theft attacks in the bot-iot dataset”. In: *2021 20th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE. 2021, pp. 807–812.

[181] Cuiju Luan and Guozhu Dong. “Experimental identification of hard data sets for classification and feature selection methods with insights on method selection”. In: *Data & Knowledge Engineering* 118 (2018), pp. 41–51.

[182] Joung Woo Ryu, Mehmed Kantardzic, and Chamila Walgampaya. “Ensemble classifier based on misclassified streaming data”. In: *Proc. of the 10th IASTED int. Conf. on artificial intelligence and applications, austria*. 2010, pp. 347–354.

[183] Valentina Timčenko and Slavko Gajin. “Machine learning based network anomaly detection for IoT environments”. In: *ICIST-2018 Conference*. 2018.

[184] Ibraheem Aljamal et al. “Hybrid intrusion detection system using machine learning techniques in cloud computing environments”. In: *2019 IEEE 17th international conference on software engineering research, management and applications (SERA)*. IEEE. 2019, pp. 84–89.

[185] Lukas Ruff et al. “A unifying review of deep and shallow anomaly detection”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 756–795.

[186] Yucheng Xiao. “Analysis of Deep Anomaly Detection Algorithms”. In: *Proceedings of the 2021 6th International Conference on Multimedia Systems and Signal Processing*. 2021, pp. 64–67.

[187] Yifan Feng et al. “An improved X-means and isolation forest based methodology for network traffic anomaly detection”. In: *Plos one* 17.1 (2022), e0263423.

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

[189] Christos Bellas et al. “Facilitating DoS Attack Detection using Unsupervised Anomaly Detection”. In: *Proceedings of the 34th International Conference on Scientific and Statistical Database Management*. 2022, pp. 1–4.

[190] Lingren Wang et al. “Anomaly detection in wireless sensor networks based on KNN”. In: *Artificial Intelligence and Security: 5th International Conference, ICAIS 2019, New York, NY, USA, July 26–28, 2019, Proceedings, Part III 5*. Springer. 2019, pp. 632–643.

[191] Xi He et al. “Detecting anomalies in distributed control systems by modeling traffic behaviors”. In: *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE. 2018, pp. 534–538.

[192] Brandon Phillips, Eric Gamess, and Sri Krishnaprasad. “An evaluation of machine learning-based anomaly detection in a SCADA system using the modbus protocol”. In: *Proceedings of the 2020 ACM Southeast Conference*. 2020, pp. 188–196.

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

[194] Franka Schuster et al. “Attack and fault detection in process control communication using unsupervised machine learning”. In: *2018 IEEE 16th international conference on industrial informatics (INDIN)*. IEEE. 2018, pp. 433–438.

[195] Xuyun Zhang et al. “LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis”. In: *2017 IEEE 33rd international conference on data engineering (ICDE)*. IEEE. 2017, pp. 983–994.

[196] Yihong Yang et al. “Astream: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment”. In: *IEEE Transactions on Network Science and Engineering* (2022).

[197] Emmanuel Aboah Boateng and JW Bruce. “Unsupervised machine learning techniques for detecting PLC process control anomalies”. In: *Journal of Cybersecurity and Privacy* 2.2 (2022), pp. 220–244.

[198] Shubin Su et al. “An efficient density-based local outlier detection approach for scattered data”. In: *IEEE Access* 7 (2018), pp. 1006–1020.

[199] Omar Alghushairy et al. “A review of local outlier factor algorithms for outlier detection in big data streams”. In: *Big Data and Cognitive Computing* 5.1 (2020), p. 1.

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

[201] Ali H Abuzaid. “Identifying density-based local outliers in medical multivariate circular data”. In: *Statistics in medicine* 39.21 (2020), pp. 2793–2798.

[202] Nelson Makau Mutua and Petr Matoušek. “Outlier Detection in Smart Grid Communication”. In: *arXiv preprint arXiv:2108.12781* (2021).

[203] Abdelkader Dairi et al. “Semi-supervised deep learning-driven anomaly detection schemes for cyber-attack detection in smart grids”. In: *Power Systems Cybersecurity: Methods, Concepts, and Best Practices*. Springer, 2023, pp. 265–295.

[204] Panagiotis Radoglou Grammatikis et al. “An anomaly detection mechanism for IEC 60870-5-104”. In: *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE. 2020, pp. 1–4.

[205] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[206] Adolfo Javier Jara Cespedes et al. “Performance Evaluation of Machine Learning Methods for Anomaly Detection in CubeSat Solar Panels”. In: *Applied Sciences* 12.17 (2022), p. 8634.

[207] Blake Anderson and David McGrew. “Identifying encrypted malware traffic with contextual flow data”. In: *Proceedings of the 2016 ACM workshop on artificial intelligence and security*. 2016, pp. 35–46.

[208] Slavica V Boštjančič Rakas, Mirjana D Stojanović, and Jasna D Marković-Petrović. “A review of research work on network-based SCADA intrusion detection systems”. In: *IEEE Access* 8 (2020), pp. 93083–93108.

[209] Nicholas Jeffrey, Qing Tan, and José R Villar. “A review of anomaly detection strategies to detect threats to cyber-physical systems”. In: *Electronics* 12.15 (2023), p. 3283.

[210] Gustavo De Carvalho Bertoli et al. “An end-to-end framework for machine learning-based network intrusion detection system”. In: *IEEE Access* 9 (2021), pp. 106790–106805.

[211] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[212] Chih-Yuan Lin, Simin Nadjm-Tehrani, and Mikael Asplund. “Timing-based anomaly detection in SCADA networks”. In: *Critical Information Infrastructures Security: 12th International Conference, CRITIS 2017, Lucca, Italy, October 8-13, 2017, Revised Selected Papers 12*. Springer. 2018, pp. 48–59.

[213] Dingde Jiang et al. “Network traffic anomaly detection based on sliding window”. In: *2011 International Conference on Electrical and Control Engineering*. IEEE. 2011, pp. 4830–4833.

[214] Ali Tekeoglu et al. “Unsupervised Time-Series based Anomaly Detection in ICS/SCADA Networks”. In: *2021 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE. 2021, pp. 1–6.

[215] Yong Peng et al. “Industrial control system fingerprinting and anomaly detection”. In: *International Conference on Critical Infrastructure Protection*. Springer. 2015, pp. 73–85.

[216] Jeroen Van Splunder. “Periodicity detection in network traffic”. In: *Technical Report, Mathematisch Instituut Universiteit Leiden* (2015).

[217] Johan Åkerberg et al. “Future industrial networks in process automation: Goals, challenges, and future directions”. In: *Applied Sciences* 11.8 (2021), p. 3345.

[218] Liu Liang et al. “Study on the Characteristics of Network Traffic based on STFT”. In: *2015 2nd International Conference on Information Science and Control Engineering*. IEEE. 2015, pp. 485–488.

[219] Charles V Wright, Fabian Monrose, and Gerald M Masson. “Using visual motifs to classify encrypted traffic”. In: *Proceedings of the 3rd international workshop on Visualization for computer security*. 2006, pp. 41–50.

[220] Neminath Hubballi and Deepanshu Goyal. “Flowsummary: Summarizing network flows for communication periodicity detection”. In: *Pattern Recognition and Machine Intelligence: 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. Proceedings 5*. Springer. 2013, pp. 695–700.

[221] Piotr Indyk, Nick Koudas, and Shanmugavelayutham Muthukrishnan. “Identifying representative trends in massive time series data sets using sketches”. In: *26th International Conference on Very Large Data Bases, VLDB 2000*. 2000, pp. 363–372.

[222] Wojciech Mazurczyk, Krzysztof Szczypiorski, and Bartosz Jankowski. “Towards steganography detection through network traffic visualisation”. In: *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*. IEEE. 2012, pp. 947–954.

[223] Abubakar Sadiq Mohammed et al. “Detection and mitigation of field flooding attacks on oil and gas critical infrastructure communication”. In: *Computers & Security* 124 (2023), p. 103007.

[224] Mauro Conti, Denis Donadel, and Federico Turrin. “A survey on industrial control system testbeds and datasets for security research”. In: *IEEE Communications Surveys & Tutorials* 23.4 (2021), pp. 2248–2294.

[225] Ivo Frazão et al. “Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process”. In: *Critical Information Infrastructures Security: 13th International Conference, CRITIS 2018, Kaunas, Lithuania, September 24-26, 2018, Revised Selected Papers 13*. Springer. 2019, pp. 230–235.

[226] Netresec. *ICS Lab: 4SICS ICS Lab PCAP File*. 2018. URL: https://www.netresec.com/?page=PCAP4SICS.

[227] Ed Skoudis, Josh Wright, and Tom Hessman. *CyberCity SANS Holiday Hack 2013 Dataset*. 2013. URL: https://www.sans.org/mlp/holiday-challenge/2013/.

[228] *Electra dataset: Anomaly detection ICS dataset*. URL: http://perception.inf.um.es/ICS-datasets/.

[229] R Sadre and G. K. Nonda. *HVAC Traces*. URL: https://github.com/gkabasele/HVAC_Traces.

[230] A. Lemay. *Modbus Dataset from CSET 2016*. URL: https://github.com/antoine-lemay/Modbus_dataset.

[231] Dale Peterson and Reid Wightman. *S4x15 ICS Village PCAP Files*. URL: https://www.netresec.com/?page=DigitalBond_S4.

[232] Jonathan Goh et al. "A dataset to support research in the design of secure water treatment systems". In: *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*. Springer. 2017, pp. 88–99.

Appendix A

Gas Wellhead Monitoring Station testbed

The testbed emulates a gas well-head monitoring station where volatile hydrocarbon fluids in gas phase (natural gas) is transported through pipelines. As gas (emulated by compressed air) is flowing through pipelines, we have to control that flow and ensure that it is flowing within a certain pressure within a certain temperature. We have to monitor the pressure, temperature, flow rate. The aim is to be able shut off operations if anything goes bad using the valves (solenoid valves).

These valves are installed at strategic points to emulate the saftey requirements of an oil and gas facility. The temperature, pressure and flow are for constantly measured and monitored to ensure they do not exceed prescribed limits. When there is too much pressure in the pipes, the overflow is sent through to the flare to avoid causing any explosion as a form of pressure relief.

The testbed was designed and installed during the course of this PhD thesis to conduct experiments, capture the communication between HMI and RTU, carry out carefully crafted attacks on the system (e.g. Denial of Service), and run analyses to check any anomaly in that communication.

Testbed hardware overview:

Solenoid Valves:

Solenoid valves come with certain specifications – either normally open (NO) or normally

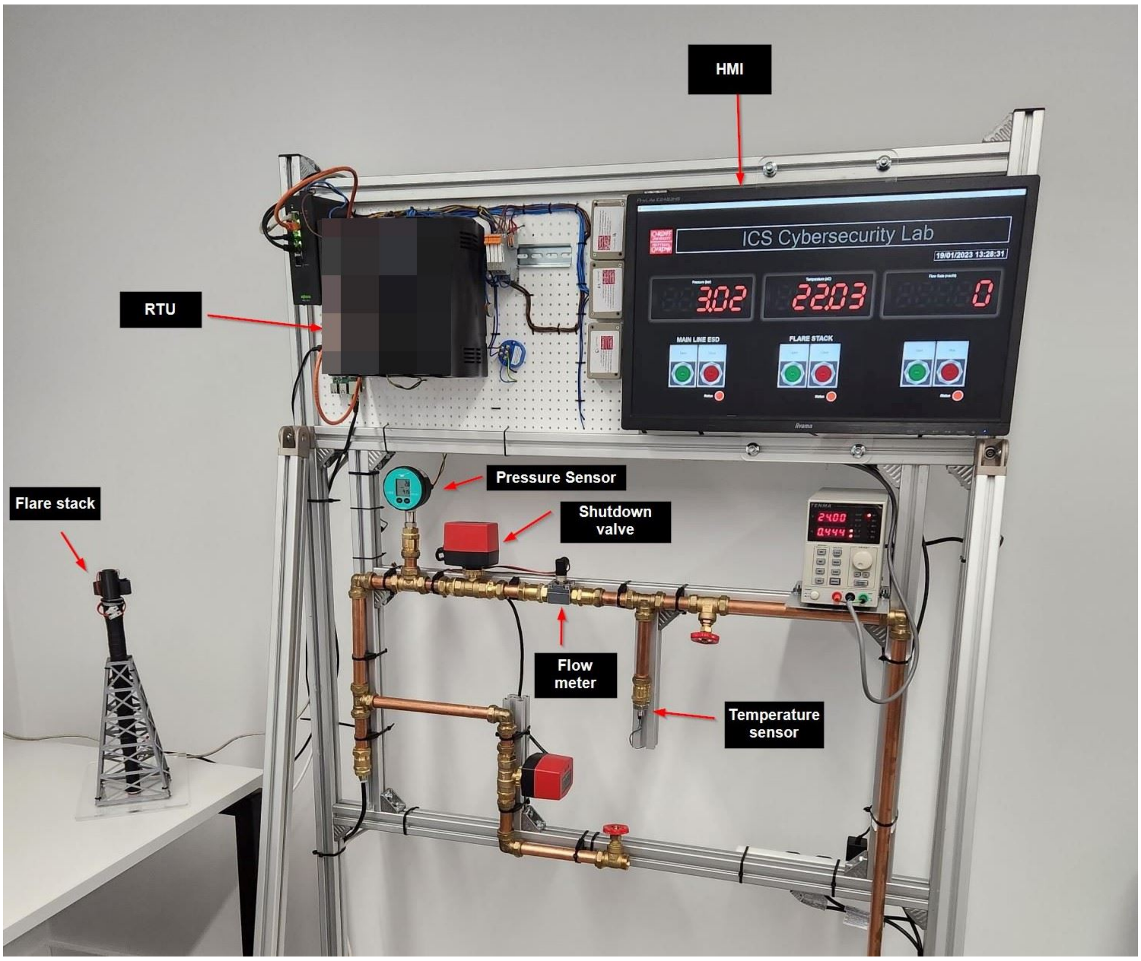


Figure A.1: Gas wellhead monitoring station testbed

closed (NC). If the valve is NO, it closes and shuts off flow when energised and vice versa when NC.

- Shutdown valve (normally open valve): allows the air to pass (flow meter on HMI would display a number greater than zero) during normal operations. When energised, it shuts off air flow and halts operations.

- Control/relief valve (normally closed valve) to direct the flow. When energised, redirects flow to the flare stack.

RTU (Modular RTU with expansion slots)

- DI (Discrete Inputs) - DI are used for discrete signals (i.e. on/off status). There are 2 programmed DIs on the testbed – status of air flow in normal operations; status of relief valve

- DO (Discrete Outputs) - Outputs to the shutdown valves (on/off status). There are 4 programmed DIs on the testbed – start/stop air flow in normal operations; start/stop relief valve
- AI (Analog Input) - AI for continuous signals such as pressure reading ranging from 0 - 20 mA. There are 3 AIs programmed on the RTU - pressure, temperature, and flow rate readings.

HMI

Displays on and off switches for mainline emergency shutdown (ESD) and flare stack. It polls data from the RTU in milliseconds using modbusTCP protocol.

On/Off Switches:

- Mainline ESD - Open: Opens the shutdown valve and allows the airflow from the air compressor (connected to the left side of the pipe). Closed: Flow meter should report zero reading (displayed on the HMI third counter - Flowrate).
- Flare Stack - Open: The flow of air changes and lights up the flare stack. The scenario is relief valve opens up to send out the excess air and that comes out through the flare stack. Closed: Flare remains non-operational.
- LED below the mainline ESD and Flare stack shows the status. Green when operational, red when non-operational. The status signals are constantly polling the controller.

Sensors

- Temperature sensor: This is a Resistance Temperature Detector or RTD sensor used to sense varying temperature ranges and is the form of a long cylindrical probe. It measures the temperature of whatever is flowing through the pipes.
- A converter (blue connector located next to RTU) is used to convert the RTD read-

ings to analogue readings (0 -20 mA signals) to enable the RTU interprete the temperature sensor measurements.

- Pressure sensor: This is a 0 - 300 bar pressure sensor installed on the main line to measure the pressure of air flowing through.
- Flow meter: measures the rate at which air flows from the air compressor through the pipes using pulse signals. A RaspberryPi is configured with a custom script to convert the pulse signals to flow rate measurements which is displayed on the HMI.

Network Switch:

- Connects the Engineering workstation (running the ROCLINK software and HMI)
- Connects the RasberryPi
- Connects attacker system

Miscellaneous Items:

- Power Supply: Supplies powers to the RTU and HMI
- Protective cases (located left side to the monitor): These are hard covers to protect the power supply cables.
- Terminal blocks (located right side to the RTU): These provide continuity to the cables.

Appendix B

Published Datasets

The datasets used in this research can be found at https://github.com/abusadiqmohd/ICS_testbed_pcaps/releases/modbusTCP

Dataset File Captures (Pcap Files):

1. Benign_capture (no attacks, clean operations depicting normal state)
2. mitm_attacks (pcap files containing Man-in-the-Middle attacks)
3. syn_flooding_attacks (pcap files containing ddos syn flooding attacks)
4. field_flooding_attacks (pcap files containing field flooding attacks)

File naming criteria:

For each file, the naming format is: [attack]_[attack duration]_[capture duration].pcapng

For instance: fieldflood_31m_1h.pcapng refers to a capture for a field flooding attack that lasted for 31 minutes over a 1 hour capture period.

Table B.1: Details of published dataset

Filename	Attack Start Index	Attack Stop Index	Attack Duration	Capture Duration	No. of Pkts
benign_capture_22hrs.pcapng	N/A	N/A	N/A	22 hrs	3,409,005
fieldflood_1h_1,2h.pcap.ng	12501	32510	1 hr	1.2 hrs	49,105
fieldflood_1h_3,8h.pcapng	414611	455338	1 hr	3.8 hrs	472,887
fieldflood_6m_1,5h.pcapng	attack 1 = 6250 attack 2 = 27409	attack 1 = 8046 attack 2 = 28862	attack 1 = 6 mins attack 2 = 5.5 mins	1.5 hrs	32,503
fieldflood_56m_1,9h.pcapng	1347	16605	56.3 mins	1.9 hrs	133,596
mitm_5,5m_2h.pcapng	298855	312681	5.5 mins	2 hrs	319,906
mitm_7m_3,45h.pcapng	375657	390500	7 mins	3.45 hrs	395,449
mitm_8m_3h.pcapng	377770	394549	8 mins	3 hrs	397,356
mitm_9m_2h.pcapng	35498	54077	9 mins	2 hrs	301,139
synflood_2m_83m.pcapng	5952	128134	2 mins	83 mins	294,397
synflood_13s_85m.pcapng	200864	216163	13 secs	85 mins	230,409