

UNIVERSITY OF PADUA

MASTER THESIS

Industrial Control Systems: Security and Privacy Analysis in Industry 4.0

Supervisor:

Prof. Mauro CONTI

Co-Supervisor:

Federico TURRIN

Candidate:

Ahmad Bashir USMAN

*A thesis submitted in fulfillment of the requirements
for the Degree of Master of Science in
ICT for Internet and Multimedia*

Department of Information Engineering

ACADEMIC YEAR 2020/2021

Declaration of Authorship

I, Ahmad Bashir USMAN, declare that this thesis titled, "Industrial Control Systems: Security and Privacy Analysis in Industry 4.0", which is submitted in fulfillment of the requirements for the Degree of Master of Science, is my own work and I confirm that the work in this thesis was mainly done in candidature for a master research degree at University of Padua, and to the best of my knowledge any material, source consulted by others is explicitly been stated. I further declared that it contains no previously submitted work for the award of degree, diploma or any other qualification to this University or other institution.

July 6th, 2021

AHMAD

“Without encryption, you and I wouldn’t be able to do our banking online. We wouldn’t be able to buy things online, because your credit cards - they’ve probably been ripped off anyway, but they would be ripped off left and right every day if there wasn’t encryption.”

Tim Cook

UNIVERSITY OF PADUA

Abstract

Department of Information Engineering

Master of Science

Industrial Control Systems: Security and Privacy Analysis in Industry 4.0

by Ahmad Bashir USMAN

In the era of industrial revolution, legacy systems continue to coexist with modern systems, constituting an integration in the same industrial control networks, this integration implies massive and heterogeneous computing embedded systems. The devices employed in these systems are using different communication protocols, operative systems, and security policies to generate entropy for the existing cyber-physical security approaches. Moreover, cybersecurity standards indicate zone segregation paradigms between Corporate zone and Control zone networks that are not or partially implemented by the organizations due to re-engineering costs. Therefore, it becomes essential not to underestimate the initial sources of potential threats. In particular, it is within the Corporate networks that the adversarial actions initiate the escalation towards disruption of control system assets.

In my work, I will provide a comprehensive analysis of an Industry 4.0, systemize the existing and the most recent work on Cyber-physical systems (CPS), analyzing the open challenges and the security issues, then I will perform data collection of the traffic from the most common Industrial Control Systems (ICS) protocols and I will perform an analysis on the encrypted traffic. In this thesis, I will consider the perspective of the security, in which I derive a taxonomy of vulnerabilities, threats and attacks associated with CPS and propose a possible countermeasure. And also I will consider the privacy perspective, where I analyze the encryption of the ICS protocols. I will establish communications between plant devices of ICS such as Human Machine Interface and Programmable Logic Controller using an open-source network simulator. Generally, the communication in ICS protocols such as Modbus, Distributed Network Protocol 3 (DNP3) and Similar protocols are presented in unencrypted format. Therefore, I will encrypt the network traffic using the Advanced Encryption Standard (AES) encryption mechanisms. Furthermore, I will infer the type of encrypted actions in the communication.

Acknowledgements

I would like to express my sincere appreciation to my supervisor Prof. Mauro Conti for his ideas, support and invaluable insight throughout the work of my thesis.

Without the support, time and guidance from my Co-supervisor Federico Turrin, I would have ever reached the end of my thesis.

I am truly grateful to all my professors at UNIPD, I have received a great deal of assistance and encouragement by their end.

I would also like to thank all the group members from the Security and Privacy research lab (SPRITZ), special thanks to my friends and colleagues at UNIPD.

Contents

Declaration of Authorship	1
Abstract	5
Acknowledgements	7
Summary	21
Outline	23
1 Industrial Revolution	25
1.1 Industry 1.0	25
1.2 Industry 2.0	26
1.3 Industry 3.0	27
1.4 Industry 4.0	28
2 State of The Art Analysis	29
2.1 Related Work	29
2.2 Background on CPS Security	30
2.3 Differences between IT and OT networks	31
2.4 IT to OT convergence	31
2.5 Vulnerabilities of ICS Protocol and Countermeasures	32
2.6 What kind of vulnerabilities generate this interconnection	33
2.7 Analysis of OT threats	34
2.8 Analysis of Purdue model	36
2.9 ISA 62443 series of standards	38
2.10 Analysis of remote command and control	39
3 Anomaly Detection Systems in ICS	41
3.1 Anomaly detection techniques in ICS and their current limitation	41
3.2 IT vs OT anomaly detection	42
3.3 Real-time vs non-real time detection	42
3.4 Analysis of existent anomaly detection for ICS in the Market	43
4 KingFisher	45
4.1 Analysis of KingFisher	45
4.2 KingFisher Architecture	46
4.3 KingFisher Limitations	47

5	Modbus Protocol	49
5.1	What is Modbus	49
5.2	Modbus layer	51
5.3	Modbus actions	51
5.4	Security issues	52
6	Message Queuing Telemetry-Transport (MQTT)	53
6.1	What is MQTT	53
6.2	MQTT Layer	55
6.3	MQTT actions	56
6.4	Security Issues	56
7	Distributed Network Protocol 3 (DNP3)	57
7.1	What is DNP3	57
7.2	DNP3 Layer	58
7.3	DNP3 actions	59
7.4	Security issues	61
8	Encrypted Traffic Generation	63
8.1	Why Traffic Encryption is Necessary	63
8.2	Experimental set up	63
8.3	Mininet	64
8.4	Mininet Scenario	64
8.5	CICFlowMeter	65
8.6	Traffic Generation	66
8.7	Data Collection	70
9	Machine Learning Techniques	71
9.1	Supervised & Unsupervised Learning	71
9.2	Unsupervised Learning	73
9.2.1	Clustering	73
9.2.2	Hierarchical Agglomerative Clustering (HAC)	73
9.3	Supervised Learning	75
9.3.1	Classification	75
9.3.2	Random Forest	75
9.3.3	Support Vector Machine (SVM)	76
9.3.4	Deep Neural Network	78
10	Data Processing & Modeling	79
10.1	Dataset	79
10.2	Data Preprocessing	80
10.3	Dynamic Time Warping (DTW)	81
10.4	Classification Metrics	81
10.5	Training & Testing modeling phase	82

11 Results & Analysis	83
11.1 Modbus	84
11.1.1 Random Forest Classifier	84
11.1.2 Support Vector Machine Classifier	85
11.1.3 Deep Neural Network Classifier	86
11.2 MQTT	86
11.2.1 Random Forest Classifier	86
11.2.2 Support Vector Machine Classifier	89
11.2.3 Deep Neural Network Classifier	90
11.3 DNP3	90
11.3.1 Random Forest Classifier	90
11.3.2 Support Vector Machine Classifier	91
11.3.3 Deep Neural Network Classifier	93
11.4 Performance Analysis and Comparison of ML Algorithms. . .	94
11.4.1 Precision Metric	94
11.4.2 Recall Metric	94
11.4.3 F1 Score	94
11.4.4 summary	96
12 Conclusion and Future work	97
Bibliography	99

List of Figures

1.1	Industries in the Great Britain in 1750 – 1850 From " <i>The Industrial Revolution</i> "(Wyatt, 2008)	25
1.2	First steam-locomotive in New York State, The deWitt Clinton, in 1831 From " <i>The Dawn of Innovation</i> "(Charles and Morris, 2012)	26
1.3	Aerospace Industry Takes Off Assembly Automation , in 2015 From (Weber, 2015)	27
1.4	Transition of the Industrial ages from " <i>literature review of the impact of the 4th industrial revolution on product design and development</i> "(Pessoa and Jauregui-Becker, 2020)	28
2.1	industroyer-schem (Cherepanov and Lipovsky, 2017).	35
2.2	Purde Model for ICS (trust, 2017).	37
2.3	ISA-62443-elements	39
4.1	Workflow & architecture of KingFisher (Bernieri, Conti, and Turrin, 2019)	46
5.1	Description of Standard serial network of Modbus with more than 247 slaves and a single master, every slave has a unique address (William L. Mostia, 2019)	49
5.2	Description of Modbus/TCP deployed on an Ethernet Network traffic with transmitted data from server to client via IP address. (William L. Mostia, 2019)	50
6.1	Process of a simple MQTT protocol utilizing a scheme of publish/subscribe	54
6.2	Description of the MQTT layer (Thiel, 2016).	55
7.1	Components of typical SCADA system connected to DNP3 (Lemaymd, 2004)	57
7.2	Description of the three DNP3 protocol and the corresponding layers of ISO/OSI model (Clarke, 2004)	58
7.3	Description of DNP3 protocol functionalities and related corresponding layers of ISO/OSI model (Clarke, 2004)	59
7.4	Simple DNP3 configuration and packet exchange (Darwish, Igbe, and Saadawi, 2016).	61
7.5	Unsolicited Message Attack (Darwish, Igbe, and Saadawi, 2016)	62
8.1	Graphical representation of the mininet scenario and the architecture of the simulation.	65

8.2	Unencrypted Modbus Traffic	66
8.3	Unencrypted MQTT Traffic	67
8.4	Unencrypted DNP3 Traffic	68
8.5	Eencrypted Modbus Traffic	69
9.1	Categories of the most commonly used Machine learning algorithms (Duc et al., 2019).	72
9.2	Hierarchical Agglomerative Clustering Structure (Zanuttigh, 2020)	73
9.3	Random Forest Classifier’s workflow, utilizing 600 weak trees learners (Chakure, 2019).	76
9.4	Simplified feedforward Neural Network scheme (Larson, 2020).	78
11.1	Modbus Confusion Matrix	84
11.2	Learning Curve for Modbus protocol	85
11.3	Precision score for MQTT protocol of random forest classifier in relation to the n estimators parameter.	87
11.4	MQTT Confusion Matrix	88
11.5	Learning Curve for MQTT protocol	89
11.6	Precision score for DNP3 protocol of random forest classifier in relation to the n estimators parameter.	91
11.7	Learning Curve for DNP3 protocol	92
11.8	DNP3 Confusion Matrix	93
11.9	Precision Performance Analysis	95
11.10	Recall Performance Analysis	95
11.11	F1 Score Performance Analysis	95

List of Tables

2.1	Stuxnet Characteristics (Chen and Abu-Nimeh, 2011).	34
3.1	Comparison of existent anomaly detection for ICS in the Market (Peterson, 2017).	43
5.1	Comparison between the ISO/OSI, TCP/IP and Modbus Layers (Omiccioli, 2017)	51
5.2	Description of Stored data in Standard Modbus with corresponding actions (Modbus. Modbus Organization, 2013).	52
6.1	Comparison between Brokers of MQTT protocol	54
11.1	User actions taken into account for each protocol	83
11.2	Modbus random forest classification metrics	84
11.3	Modbus SVM classification metrics	85
11.4	Modbus Deep Neural Network classification metrics	86
11.5	MQTT random forest classification metrics	87
11.6	MQTT Support Vector Machine classification metrics	89
11.7	MQTT Deep Neural Network classification metrics	90
11.8	DNP3 Random Forest classification metrics	90
11.9	DNP3 Support Vector Machine classification metrics	91
11.10	DNP3 Deep Neural Network classification metrics	93
11.11	Precision Average Results for Machine Learning algorithms	94
11.12	Recall Average Results for Machine Learning algorithms	94
11.13	F1 Score Average Results for Machine Learning algorithms	94

List of Abbreviations

AES	Advanced Encryption Standard
ANN	Artificial Neural Networks
CIC	Canadian Institute for Cybersecurity
CIP	Common Industrial Protocol
CN	Correlation Node
CNN	Convolutional Neural Network
COO	Chief Operating Officer
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access with Collision-Detection
DCS	Distributed Control System
DNN	Deep Neural Network
DNP3	Distributed Network Protocol 3
DNP3-SA	Distributed Network Protocol 3 Secure Authentication
DoS	Denial of Service
DTW	Dynamic Time Warping
HAC	Hierarchical Agglomerative Clustering
HIPS	Host Intrusion Prevention System
I1.0	Industry 1.0
I2.0	Industry 2.0
I3.0	Industry 3.0
I4.0	Industry 4.0
ICS	Industrial Control Systems
IDPS	Intrusion Detection and Prevention Systems
IoT	Internet of Things
IR1	First Industrial Revolution
IR2	Second Industrial Revolution
IR3	Third Industrial Revolution
IR4	Fourth Industrial Revolution
IT	Information Technology
LRC	Longitudinal Redundancy Check
M2M	Machine-To-Machine
MFCM	Modbus Function Code Modification
MITM	Man In The Middle
MPB	Modification of Physical Behavior
MQTT	Message Queuing Telemetry Transport
NIDS	Network-based Intrusion Detection System
NN	Neural Network
OASIS	Organization for the Advancement of Structured-Information Standards
PLC	Programmable Logic Controller
RBF	Radial Basis Function

RF	Random Forest
RSA	Ron Rivest Adi Shamir Leonard Adleman
SCADA	Supervisory Control and Data Acquisition
SIS	Safety Instrumented System
SVM	Support Vector Machine
TCP	Transmission Control Protocol
VAEs	Variational-Auto Encoders

Dedicated
To my Mother and Father
who always encouraged me to take the advantage of
every adventure.
To my brothers and sisters for always loving and
supporting me.
To myself for believing in me.

Summary

In 2011, the term Industry 4.0 (also known as I4.0) was first publicly announced at the Hannover Messe in order to promote the manufacturing computerization of the industrial sector in Germany (Kagermann, Wahlster, and Helbig, 2013). The term Industrial revolution is an open debate and interpreted in different ways by different researchers and it is freighted with several meanings in many articles (Clark, 2010). Not only that the Industrial revolution changes the technical advancements in technology, but it also changes the human capital and the way they are actively creating creative things. Fundamentally, the new innovation technologies have completely changed the lifestyle of human beings and the working conditions from the very first industrial revolution to what we currently found ourselves now in Industry 4.0.

In this thesis, I firstly introduce briefly the Industrial revolution, and their transition from the first, second third, highlighting their difference, and dive deep into the fourth industrial revolution, taking into account the common security lacks, focusing on the vulnerabilities, threat and the attacks, while also considering the convergence of the IT and OT networks, with a special analysis on anomaly detection systems.

Having demonstrated a comprehensive analysis of Industry 4.0 and the security concerns associated with it, I will practice privacy with encryption in the industrial control systems protocols. Recalling that the communication in SCADA and ICS are usually insecure, we believe that encryption in such communication will make the privacy possible, thus yielding an extra security in the Industrial Control Systems.

Outline

The ultimate goal that I am trying to achieve in this thesis is to implement an encryption in the industrial control network traffic, and conduct an analysis of the encrypted traffic. Bearing in mind that the ICS protocols normally are not encrypted. Such behavior can leak information about the communication, as a result, an attacker can easily eavesdrop the communication and infer the type of the protocol used, the topologies and other sensitive information. In the literature, similar work has been done previously in different fields such as the Internet of Things (IoT), side-channel and Android mobile applications. To the best of my knowledge, this is the first work that analyzes the encrypted network traffic of the Industrial Control System's protocols.

The overall steps and the expected outcomes of this thesis can be visualized as following items:

- Background on Industry 4.0 security and in particular ICS security.
- Convergence and analysis of IT/OT networks: legacy systems and Industrial Internet of Things (IIoT).
- Analysis of OT threats: Stuxnet; TRISIS; Maroochi; Industroyer.
- Analysis of the Purdue model, ISA 62443 series of standards, IT/OT network segregation
- Analysis of IT cyber vulnerabilities conceived for remote C&C
- Analysis of Intrusion & Anomaly detection in Cyber-Physical Systems.
- Analysis of the Industrial Control Systems protocols
- Data collection of the traffic from the most common ICS protocols.
- Implementation of encryption on the ICS protocols.
- Analysis of the encrypted network traffic using machine learning.

Chapter 1

Industrial Revolution

1.1 Industry 1.0

In the first period of industrialization, the transition to machine and factory systems from the world, full of artisan manufacture is referred to as an industrial revolution. In the early period of the eighteenth century, the transition began in Britain (CRAFTS, 2011). This story originally starts in a small island in Britain. In this century, people were using the available trees in order to build houses and produce ships, also cooking food and heating purposes.

The new transition of manufacturing processes in the United Kingdom, United State and Europe was brought by virtue of the Industrial revolution. Among this substance transition including but not limited to, instead of using the traditional and hand-production methods, now people are deploying machines. This progress extended to iron production and chemical manufacturing. The use of water power and steam power has started to increase, the mechanized-factory system is rising as well as the development in machine tools. These revolutions lead to unexpected growth of the population.

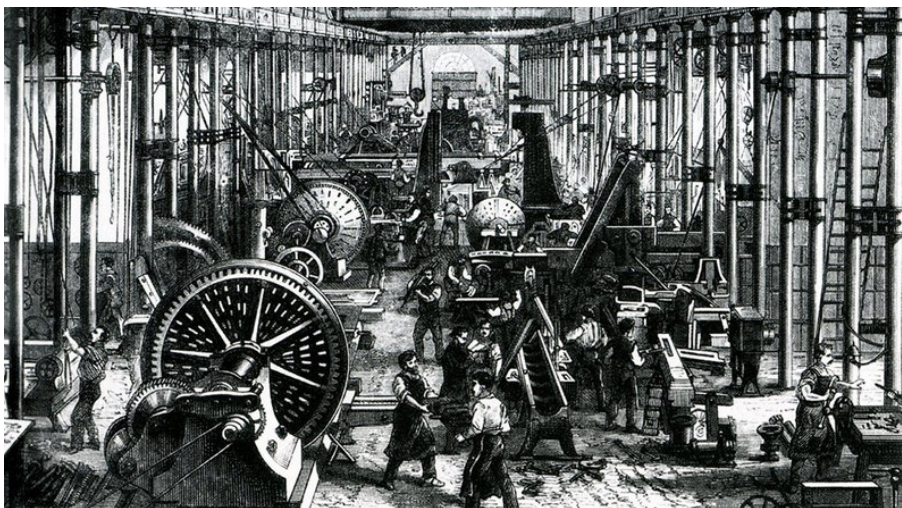


FIGURE 1.1: Industries in the Great Britain in 1750 – 1850 From
" *The Industrial Revolution*" (Wyatt, 2008)

In the period of Industry 1.0, human activities were changed, it encouraged people to make a transition from traditional to revolution, allowing them to focus on industrial society from the agricultural environment. During that era, there were only one dimension for the demand of industrial products, it called the product volume (Yin, Stecke, and D. Li, 2018). In Industry 1.0, the demands were bigger than the supplies, as a result, the output from the industrial products was not sufficiently enough to satisfy the required demands from society. One of the best books that cover various aspects of Industry 1.0 is a book called *wealth of nation* by Adam Smith (Smith and Krueger, 2003).

1.2 Industry 2.0

The appearance of the Second Industrial Revolution (IR2) started in the period between the 1860-1914 (Gordon, 2000). This revolution, also known as the American Industrial Revolution, is due to the creation of a huge amount of utilities and the invention of new technologies which include internal combustion engines, electricity, petroleum, alloys, chemical industries and other chemicals. The author Gordon (Gordon, 2000) also includes the electrical communication technologies such as telephone, telegraph and radio for this revolution. Researchers in (Atkeson and Kehoe, 2001) mentioned that this period was the moment in which the innovations and inventions concentrated on steel and iron, electricity railroads and chemicals were highly science-based. *The age of Synergy* was the proposed name by Vaclav Smil the policy analyst and a Canadian scientist during this period of innovations and inventions as stated in his reviewed article and was originally authored by Petrie Ian, talking about *Creating the 21th century* (Petrie, 2007).

The second Industrial revolution is the period of creation of the current transportation, industrial economy, development of steam power and the generation of communication.

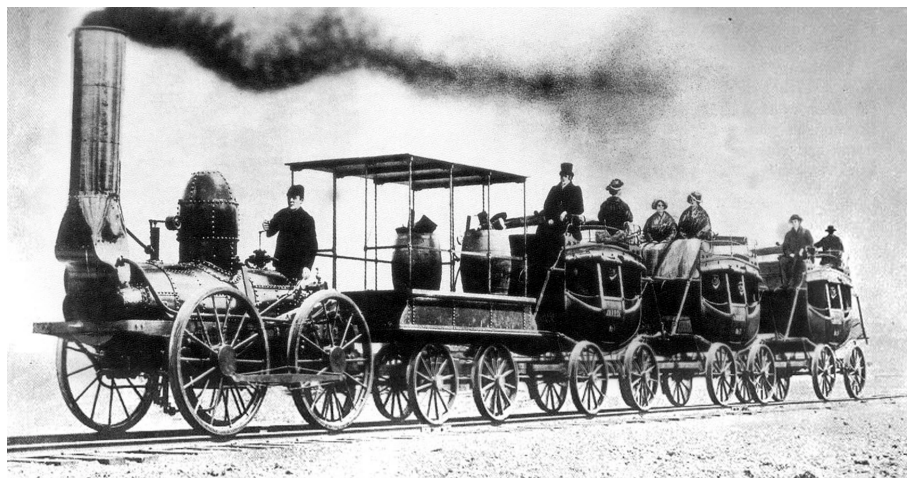


FIGURE 1.2: First steam-locomotive in New York State, The de-Witt Clinton, in 1831 From "*The Dawn of Innovation*" (Charles and Morris, 2012)

1.3 Industry 3.0

The beginning of Industry 3.0 was noticed in the 1980s till today. The start of technological innovations was realized during this period and it has a big impact in the world of the electronics industry. Changes such as moving to digital from analog and to modular from integral. The architecture of most electronics was accompanied by decreasing product life cycles. The case in Japan was almost six months of the average life-cycle in the production of electronics (Yokoi, 2014). The creation of the Third Industrial Revolution helped the current century by creating more new businesses in the market, and many more opportunities for the current sustainable global economy. Again, the Third Industrial revolution has changed not only how we communicate but also the way we perceive and deeply define our insight to the universe.

Back in 2012, Jeremy Rifkin mentioned that the last of the greatest inventions of the Industrial Revolutions is the Third Industrial Revolution and that the foundational infrastructure was carried and helped for collaborative age emergence (Rifkin, 2012).

CD-ROM was one of the potential electronic storage media that were invented in this era and it tool has supported the revolution of Information Technology. In the InforTech market, researchers made an announcement about the CD-ROM, in which they announced that more than 155% the growth of the CD readers was achieved in 1993 (Fitzsimmons, 1994)



FIGURE 1.3: Aerospace Industry Takes Off Assembly Automation , in 2015 From (Weber, 2015)

The world experienced heavy investment in the industrial systems, the aim of the third industrial revolution was to produce more flexible systems, boost productivity, improve quality and reduce cost. In Aerospace automation, as in the figure 1.3, sectors of commercial airlines are also leading a charge by creating new airplanes, adopting robots and similar technological production to build more capacity constraints and to solve their problems.

1.4 Industry 4.0

In 2011, the term Industry 4.0 (also known as I4.0) was first publicly announced at the Hannover Messe in order to promote the manufacturing computerization of the industrial sector in Germany (Kagermann, Wahlster, and Helbig, 2013). The term Industrial revolution is an open debate and interpreted in different ways by different researchers and it is freighted with several meanings in many articles (Clark, 2010). Not only that the Industrial revolution changes the technical advancements in technology, but it also changes the human capital and the way they are productive in creating creative things. Fundamentally, the new innovation technologies have completely changed the lifestyle of human beings and the working conditions from the very first industrial revolution to what we currently found ourselves in Industry 4.0.




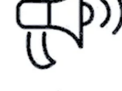



	1 st 1784	2nd 1870	3rd 1969	4th 2010
Technology 	Water power, steam power	Assembly line, electrical power	Computational power	Internet Cyber physical systems
Production 	Mechanization	Mass production	Automation	Enterprise Resource Planning Autonomy
Sales 	Cost reduction	Volume	Volume + quality	Smaller batches, wider variety Mass individualization
Marketing 	Offer creates the demand	Differentiation attributes (company push)	Satisfy the customer (customer pull)	Customer Relationship Management Sensing and responding
Product 	Simple product	Complex product	Complex system	Complex value chain Changeable product-service system
Design Dev. 	Empiric Product Development	New Product Development	Integrated Product Development	Lean Product Development Continuous Product Development
Environ. 	--	Labor rights	Pollution	Quality, Health, Safety and Environment Resources consumption

FIGURE 1.4: Transition of the Industrial ages from "literature review of the impact of the 4th industrial revolution on product design and development" (Pessoa and Jauregui-Becker, 2020)

The diagram in figure 1.4 illustrates the transition of the industrial revolution, starting from the first to current era of revolution, and compares the major differences brought into the field of technology, production, sales & marketing, design development and environment.

Chapter 2

State of The Art Analysis

2.1 Related Work

M. Conti et al (Conti, L. Mancini, et al., 2015) introduced a framework that allows them to simulate and understand vividly how the encrypted network traffic transmitted, and allows them to analyze, specify and categorize what kind of activity has been carried out by the user, taking advantage of some mobile application (e.g facebook, twitter and gmail). In their demonstration, they assumed that regardless of deploying the two well-known transmission protocols so-called SSL/TLS, it is possible for the eavesdropper to invade the privacy of the users with an effective tool in the approach provided in their traffic analysis, which proves the weakness of SSL/TLS.

R. Atterer et al. (Atterer, Wnuk, and Schmidt, 2006) built a transparent and complex solution for tracking the user activity on the internet. Their approach makes the tracking of the user activity possible in a web application, which we already knew that web applications deploy JavaScript heavily on its pages. In the implementation, they performed a small test with twelve participants that are browsing the internet through computers, each assigned with a couple of tasks which are hardly tractable providing two encrypted websites.

M. Liberatore et al. (Liberatore and Levine, 2006) proposed two machine learning algorithms to examine and analyze the effectiveness of actions which allows them to de-anonymize encrypted HTTP protocols. The first method is based on Jaccard's coefficient while the other is naive-Bayes. They have demonstrated that the observer has the ability to infer the component of the HTTP streams that is encrypted by taking advantage of the collected profile in the library: this can be done either before the encryption of the streams or after.

M. Conti et al. (Conti, L. V. Mancini, et al., 2016) again, this paper contains the same concepts and methodologies applied as in the preview paper[4]. But this paper is slightly different in terms of the amount of the dataset, where they used 7 different applications downloaded and installed from the official android market. The seven applications are: Facebook v.3.8, Twitter v.4.1.10, Gmail v.4.7.2, gplus v.5.3.0.9103405, Tumblr v.3.8.6., Dropbox v.2.4.9.00, and Evernote v.7.0.2. They provided a good performance in terms of precision, recall and F-scores.

In this thesis, I will simulate the procedures performed and provide the methods used to achieve the results as in (Conti, L. Mancini, et al., 2015) and (Conti, L. V. Mancini, et al., 2016) but only on the ICS protocols, encrypting and generating the traffic using mininet and analyzing it with CFlowMeter. And by training a machine learning algorithm with data that has been traffically encrypted, build the model and finally predict and classify the message sent or action performed by the user.

2.2 Background on CPS Security

Cyber-Physical Systems (CPSs) are novel digital technology that Nowadays arises attention among both developers in the industry and researchers in academia. CPSs can be considered as a system of systems and can be defined when the three acronyms are completely resolved: Cyber is the control part and the intelligence: physical relates to the physical world: and systems which concern the computation in processing information to make decisions, communication and collaboration in exchanging the data. These systems are increasing in number, therefore, several application domains with the current developments in technology for CPSs. Example including but not limited to; smart grid applications, medical devices, vehicular or smart cars and industrial control systems. The cost of failure and error of these systems can cause disaster effects in the physical world.

The growth of CPSs has increased exponentially and unprecedentedly in the last decades (Atat et al., 2018), and our lives rely highly on computrazid networked environments, taking advantage of these physical systems in many different ways, starting from the simplest devices that we might be able to hold in our hands, to the most complex and sophisticated systems that we might see in critical infrastructure. The emergence of CPSs brought unprecedented integration and interaction between systems and human-being. They also generated severe negative consequences due to the careless misuse, administration or unexpected attack due to the lack of security compliance in place.

Cyber attacks are mostly appeared in relation with CPSs and their vulnerabilities of the computational communication systems (Ma, Rao, and Yau, 2011). We can believe this since the security issues on the systems can usually be associated with the CPSs. Take an example of a malicious user who having a partial or full control over a computer system of medical devices, water pumps and electric gas valves, he can take advantage of this privilege to intentionally influence the physical world, lead to serious damage to the environment and put human-beings' lives at risk. This is a good example for the universe to consider security countermeasures into account, and is trustworthy to put a considerable amount of effort, money and time when designing and implementing CPSs security.

In defining CPSs, authors in (Humayed et al., 2017) defined CPSs as systems that are used for controlling and monitoring the Physical world. Gollmann and Krotofil described CPSs as a systems that are made up of of IT systems and all are embedded within a specific applications (Gollmann and

Krotofil, 2016). While on the other hands CPSs attacks are the attacks that can physically influence propagations (Yampolskiy et al., 2012). Generally, any action in cyber-space contains some sort of impact in the physical world, whether the designated system can be categorized as cyber physical or not, but it may not be classified as a potential source of damage. For example, in Information Technology (IT) and Operational Technology OT, when transmitting information through wire or wireless, the components such as monitor, hard drive, printers and so on, can have some form of physical influence propagations, but not necessarily a source of threat. In this thesis, we will mainly be focusing on Industrial control systems (ICS). Since in the networked-environment, ICSs are facing severe cyber security challenges, risk and threats (Zhou et al., 2020).

2.3 Differences between IT and OT networks

Until recently, the "terms" of Information Technology and Operational technology were distinguished both organizationally and technically.

It's clear that the term IT is widely seen and used in many different fields and not only it is familiar to everyone, but also people have a satisfactory idea of what IT is about. While on the other hand the term IT could be less familiar and might raise an alarm to the beg the question what OT is really means. The term OT may sound recent, however, It might be impossible to demonstrate the current evaluations and developments of IT in the industry without explicitly or implicitly talking about the convergence of IT and operational technology. As a result, It is important to distinguish between IT and Operational Technology.

The term Operational Technology mainly focuses on the monitoring and controlling the industrial-process assets and the industrial equipment. While in brief, the term IT is the way that enterprise and business systems deliver, store and process information.

2.4 IT to OT convergence

The transformation of digital technology in companies and in particular the industrial sector, is continuing to force them to take this paradigm into account and reconsider conducting convergence projects and bring together these two terms.

The advantages of controlling both IT and OT convergence can make an improvement in several areas, such as enhancement of information for decision making, optimization of business processes, lowering risk, reducing costs and shortening the project schedule.

Based on my research, i conclude the convergence of IT and OT as follows:

- The scope in **Information Technology** is general and the domain fall to support the enterprise applications and manage office employees. Typically the person in charge is the Chief Information Officer (CIO).

IT follows the interconnected application system approach with open and standards-base architectural model, examples include ERP, BI and CRM.

- The scope in **Operational Technology** is specialized and the domain fall to support industrial and environmental control and monitoring. Typically the person in charge is the Chief Operating Officer (COO). OT follows the standalone application system approach with closed and proprietary architectural model, examples include SCADA, MES and EMS.

2.5 Vulnerabilities of ICS Protocol and Countermeasures

In previous studies, O. Nyasore et al. (Nyasore et al., 2020) introduced vulnerabilities associated with **Modbus/TCP**, a well-known and a legacy protocol that has been widely used in electronic devices particularly in industrial control systems such as refinery control. In this paper, they developed Snort, Bro and Suricata intrusion-detection and prevention-systems (IDPS) taking advantage of deep packet inspection as a countermeasure to mitigate the denial of service, command injection and other malicious activities that may arise in Modbus/TCP in the industrial control systems. In the experiment they illustrated the results using Pingplotter which allows them to measure and compare different performance of the three IDPSs. The drawback of this experiment is the latency in predicting the results, therefore, it is possible to easily capture the attack on Modbus/TCP in real-time even with the security countermeasures in place.

M. Marian et al. (Marian et al., 2019) proposed a solution which is architecturally designed to secure industrial control systems and specifically issues related to Supervisory Control and Data Acquisition (SCADA). The authors in this paper restricted the solution on **Distributed-Network Protocol (DNP3)**. DNP3 was introduced in 1993 and consists of communication protocols which operate at data link, transport and application layers and is mainly used in electricity, water and other data acquisition systems, primarily based in Canada and U.S but later sparsely distributed in Europe (Colantes and Padilla, 2015). The four threats that DNP3 overcoming are eavesdropping and spooning, replay and modification. By modifying DNP3, the authors in (Marian et al., 2019) will guarantee data origin authentication and data integrity in electronic signature which behave as remote-terminal-units. The issue with this experiment is that the proposed system is still under development, therefore, we can't predict the efficiency of the system so as to assume the problem has been addressed.

Guilherme Serpa Sestito et al. (Sestito et al., 2014) proposed a deep learning technique to diagnose traffic in the industrial control systems. The implementation of this algorithm was focusing primarily on Artificial Neural Networks(ANN), and predicts the abnormal signals transmitted in **PROFIBUS-DP** protocol. This protocol is another legacy protocol that was introduced by the institutional research department in German back in 1989 (Collantes and Padilla, 2015).

The authors in (Sestito et al., 2014) provided a waveform sample signal feature which allows the ANN to indicate possible issues associated with it. The experiment was done in an environment using real data measured in a laboratory. The data was splitted into training sets with 70% of an input, validation and testing sets with remaining 30%. The experiment showed satisfactory results, but since the signals have different waveform in the networks configurations, different implementation on ANN should be considered when generalizing the result. Other important protocols in the industrial control systems that are widely used in Europe and the world are: **Common Industrial Protocol (CIP), Profitnet, OPC, PowerLing Ethernet and EtherCAT**. Each of these protocols are functioning in the OSI or TCP/IP model.

2.6 What kind of vulnerabilities generate this interconnection

In order to understand this vividly we have to differentiate between the types of the vulnerabilities that can arise in IT/OT based on several characteristics, such as what caused the vulnerability, where it exists and how it can be compromised.

- **Operating System Vulnerabilities:** Since today's operating systems include many functionality and are very complex, it's very difficult for a developer to design and develop softwares without an error within the operating system.

Examples of this type include: **Remote Code Execution**, the attacker can modify or execute commands remotely: **Denial of Service (DoS)** degrading or denying services to a victim: **Privilege Escalation** having access to the root and gain full permission without authorization.

- **Network Vulnerabilities :** These types mainly are the problems associated with the network's software and hardware that makes it possible to expose weakness of the network, therefore, make it possible to be attacked by malicious outside parties.

Examples include default Wi-Fi Routers to the access point, zero configuration or poorly-configured and weak implementation of firewalls.

- **Process Vulnerabilities** Example of this type of vulnerability is weak passwords creation. Implementing easily guessable passwords can easily be compromised by using brute force or similar attacks.

Aspect	Stuxnet	Common malware
Targeting	Extremely selective	Indiscriminate
Type of target	Industrial control systems	Computers
Size	500 Kbytes	Less than 1 Mbyte
Probable initial infection vector	Removable flash drive	Internet and other networks
Exploits	Four zero-days	Possibly one zero-day

TABLE 2.1: Stuxnet Characteristics (Chen and Abu-Nimeh, 2011).

- **Human Vulnerabilities** The weakest side in Information Technology and Operational Technology is basically the human element. Employees without proper training awareness in cybersecurity could easily create compromisable access points, expose very sensitive information to intruders and cause unintentionally serious damage to the systems. **Social Engineering** is the most vulnerable way that attackers take advantage of to lure the users and gain the desired goal.

2.7 Analysis of OT threats

Stuxnet is the first weapon found in the war of cyber-physical systems. At the moment stuxnet was created, it was certainly the most remarkable malware discovered in space and has the ability to attack cyber-physical systems in the military. It's main purpose was to target the facilities and the industrial control systems, and also the facilities in Natanz targeting specifically the uranium enrichment. In comparison to the previous malwares in 2010 that are focusing on targeting computers, Stuxnet was targeting Industrial control systems, and under certain situations it placed its payload to the ICSs, therefore, it was considered to be the greatest of its time in terms of complexity. The table 2.1 provides an overview of the Stuxnet main characteristics in comparison to the other malware.

Neither country admitted to holding responsibility for the damage caused by Stuxnet in Iran, but authors in (Nourian and Madnick, 2018) claimed that VirusBlockAda company was the first to discover Stuxnet malware in the mid year of 2010.

After successfully Stuxnet sabotaged the nuclear program in Iran, a new generation of war began. **TRISIS**, also known as HatMan, was one in the front line among the new malware and destructive paradigms that is capable of disrupting and altering the tasks of Safety-Instrumented-Systems(SIS).

It's worth to mention SIS is seen in Chemicals, Gas, Oil, Utilities and other related instruments, to support and provide security countermeasure the ICSs and shutdown safely the ongoing process in case of anomaly detection. TRISIS was discovered by Dragos in 2017, when they investigated ICS

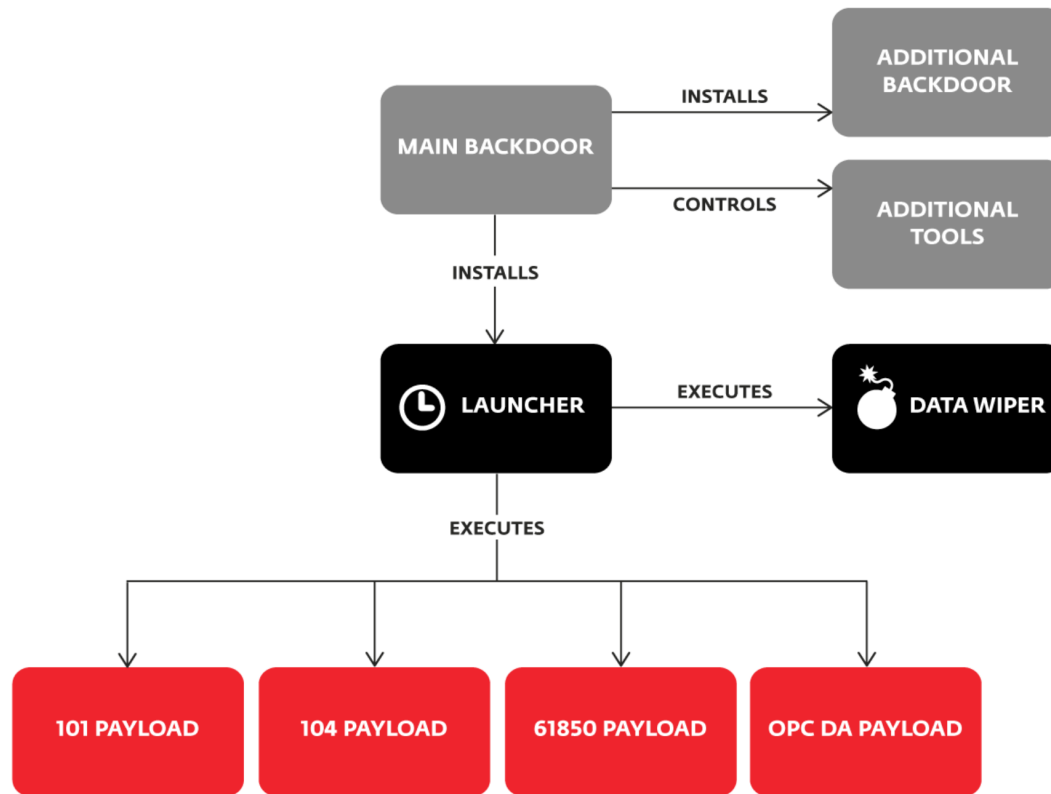


FIGURE 2.1: industroyer-schem (Cherepanov and Lipovsky, 2017).

tailored malware that attacked Triconex safety-system in the Middle East. Specifically in Saudi Arabia (Geiger et al., 2020).

Another malware that is involved in malicious activities is the so-called **Maroochy** malware. The name came after an attack occurred in Australia in the Queensland Area. It is an attractive area, visited by many tourists, known with its beauty in nature and multiple parks and water canals.

The Council in the city implemented complicated SCADA-based critical infrastructure to manage more than 140 pumping stations and 880km of sewers. Few months later, the operators noticed that there was no response for remote commands, pumps didn't run properly, reports to the central control were missed and often lost communication to the central control.

Even though the operators re-installed the software and verified thoroughly the system and seems everything working as needed, this action didn't resolve the issue (Kawano and Mustard, 2006). After several investigations for more than two months, it turns out that the attack was performed by an insider who has been rejected to be employed. The attacker was later sentenced to two years in jail for the serious damage he caused.

Final giant malware I would like to discuss in this part is the **Industroyer**, also known as **Crashoverride**. This malware is designed purposely to target electrical grids and disrupt the functionality of Industrial Control Systems. It is revealed by ESET (Cherepanov and Lipovsky, 2017) after it shut the Ukrainian power grid down for one hour.

The architecture of Industroyer as shown in figure above 2.1 provide support to four different types of protocols in industrial control systems, OLE process-control Data-Access, IEC-61850, IEC-60870 version 101, and IEC-60870 version 104 that can transmit over TCP/IP. The payload of these protocols proceeds by mapping specific networks, further sending commands to the targeted Industrial Control System. The authors of this malware not only took advantage of these protocols, but they also implement Denial of service attacks.

Other components associated with the industroyer include **Backdoor**, it's the main backdoor, where the entire components of the tool has been controlled by the attackers.

The backdoor connects to remote commands and manages the servers so as to enable the attackers to exchange, transmit and perform malicious activities over the internet. In case of failure of the main backdoor, **Additional Backdoor** is in place as a backup door in order to regain the access to the victim and provide persistency.

Another component of the industroyer is the **Launcher component** which contains the timestamp of two the activated date which is ahead of the actual attack.

Finally the **Data Wiper Component**, after successfully exploiting the systems, the data wiper will erase its crucial registry-keys and make it extremely difficult to recover. This proves that the attackers of this malware are very well funded, dedicated and expert in the industrial control systems.

2.8 Analysis of Purdue model

In order to properly apply security in ICSs and Critical Infrastructure (CI) environments, it's a vital to have detail knowlaged and be aware of the entire network components in the IT and OT. This will allow us to holistically make an investigation of the process and the system in each part, analyze vulnerabilities, risks and threats and recommend possible solutions. To be able to do so, the Purdue model and ISA 62443 standard are implemented to control the system and secure its components properly.

The Purdue model (trust, 2017) was first developed by Theodore J. Williams next to other members at Purdue university in the 1990s. Purdue Enterprise Reference Architecture (PERA) model as the name suggests it's maily for enterprises, proves to be a good example defining and distinguishing the five layers/level of critical infrastructure which is deployed in production-lines and provide a better approach to apply security. The implementation of the PERA will resolved the air-wall between ICSs and the main components presented in the IT and the OT. The overview of the Purdue model can be seen in figure 2.2

- **Level-4/5 – Enterprise:** The enterprise is the level where day-to-day to activities and primary business functions are performs. As of today, It's the level of Information Technology that supply orchestrates operation and business direction. This is the most important and most

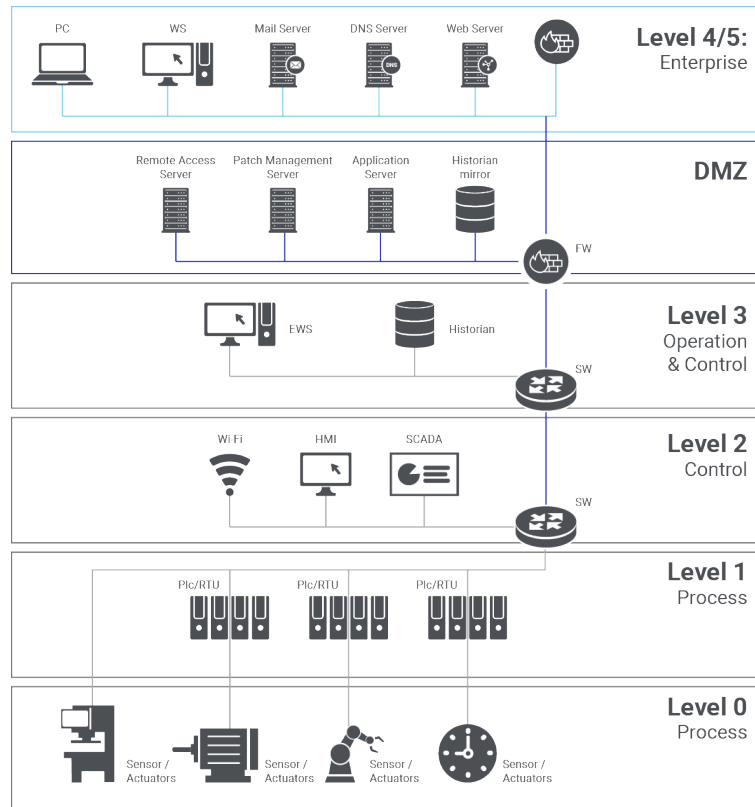


FIGURE 2.2: Purde Model for ICS (trust, 2017).

sensitive layer where disruptions of any kind can cause the enterprise huge amount of many.

- **Demilitarized zone (DMZ):** The region where IT and OT is perfectly converge is the so-called Demilitarized zone. This area possess most of the security softwares, such as proxies and firewalls. Many ICSs are suffer from the lack of security at this area.
- **Level-3 - Operation and Control:** Level 3 is the level where on manufacturing ground, the management of the workflow prudcion is taking place. Windows, Linux, Mac ios and other customized operating systems can be seen here in practice recording data, performing batch management and managing several operations. The term manufacturing execution systems(MES) and manufacturing-operations management systems(MOMS) are the names of the systems that has being manufactured at this level where the record of data are stored in the database. There is a designated backhaul network when communicating with manufacturing layer and the enterprise layer to the data-center. Interruptions at this level during execution can also cause the system to shutdown for several days which, as a result, lead to a serious loss of revenue.

- **Level-2 - Control Systems:** SCADA software has the ability to control ICSs from the actual location of the of the plants to a very long distance where the operator usually monitoring the systems.

While Programmable logic controllers(PLCs) and the Distributed control systems(DCS) are oftely implemented in the plant. Both PLCs and DCS are used for basic monitoring and control, while the transmission and aggregation of the data to upstream to level 3 is done by the help of SCADA in connection with the Human Machine Interface (HMI). The issue is that the PLCs do not posses monitors and keyboard, therefore, in order to log into SCADA systems, the operator should use Remote TerminalUnits (RTUs). The communication over modbus and dnp3 protocols along with other strategies and devices at this layer can provide assistant to bolster the security.

- **Level-1 Intelligent devices:** Manipulation and sensing of the physical-processes with the corresponding process actuators, sensors and the related instrumentation are collapse at this level. Taking advantage of of cellular networks as an example, it is possible to efficienctly drive sensors directly in communication via cloud with the dedicated monitoring software.
- **Level-0 – Physical process:** The components of the physical processes are defined at this level

2.9 ISA 62443 series of standards

ISA 62443 series was defined by the International Society of Automation committee hoping to gather experts in the Cyber-physical systems field in order to address issues related to ICSs. Their primary goal is to enhance the safety, confidentiality, integrity and availability for the system installed in the ICSs and provide procuracy criteria for the implementation of the intended ICSs. The series also aimed to enhance the electronic security and support to verify and resolve vulnerabilities, which may reduce the risk to compromise sensitive information and cause the failure of the ongoing ICSs process. The components of the series can be seen in figure 2.3, the elements are categorized into four groups with the description associated with each element.

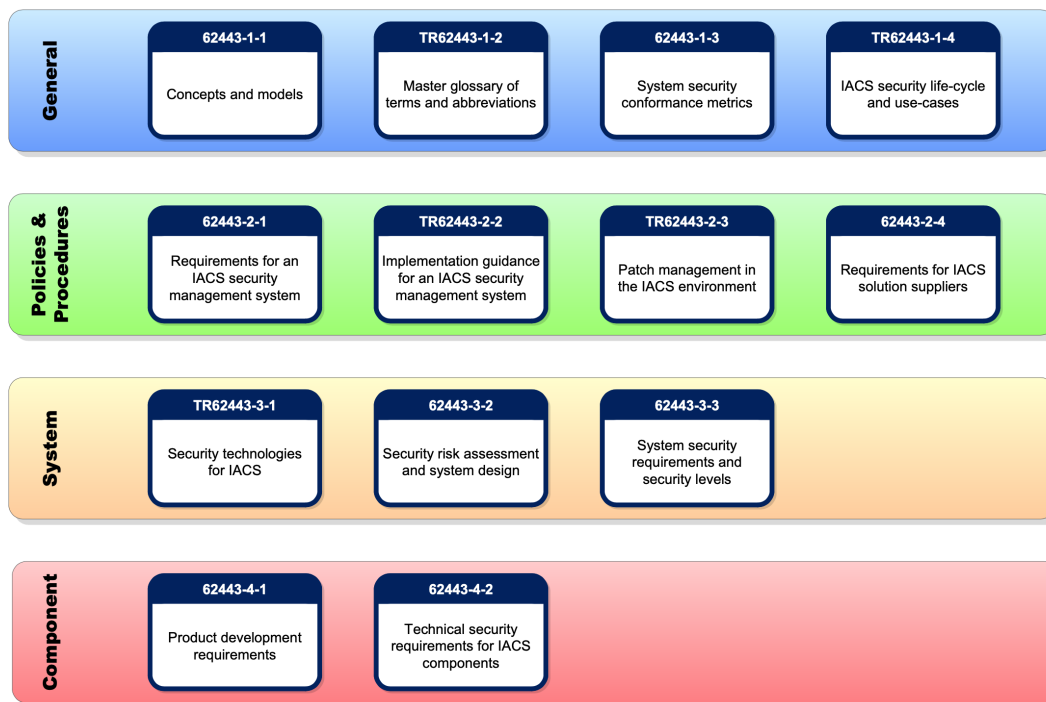


FIGURE 2.3: ISA-62443-elements
(Franceschetti et al., 2019)

2.10 Analysis of remote command and control

Before digging into demonstrations of various cyber vulnerabilities that could possibly be leveraged from remote command and control, we need to understand the objectives of such an action.

Command and Control (C&C) also known as **C2 servers**, consists of centralized computer systems that are capable of executing commands remotely and receiving outputs of the targeted system. It's possible that this action can also be controlled by malicious users, compromise the system, take advantage of several vulnerabilities and launch a DDoS for example or receive data from a compromised machine remotely.

One of the most common vulnerabilities conceived for remote command and control is the so called **Distributed Denial of Service(DDoS)**. DDoS attack is the action of overwhelming traffic by executing commands to the target machine by attackers from multiple sources in an attempt to seize its functionality rendering it unavailable.

Watkins et al. overviewed Dirt Jumper family (DJF) which is a toolkit of DDoS, aiming to search for a certain vulnerabilities hopping to stop in-progress attacks of DDoS by fuzzing the DJF manually on C&Cs. Their mitigation campaign on DDoS focuses on three vulnerabilities of DJF botnes. The first is "Weak HTTP-Login-Authentication" vulnerability which fetches the passwords continuously from the dictionary that contains all the stored passwords. The second one is the "No-Boat-Registration-Authentication" and the final one is "No Bot Input Sanitization" which they desire to investigate when considering the work for the future (Watkins et al., 2015).

Another common remote attack is the **Domain Name System (DNS) poisoning**, also known as **DNS spoofing**, is an attack that aim to exploit and take advantage of this vulnerabilities by tricking the DNS server to believe the illegitimate data is authentic and legitimate then deviate and forward the legitimate traffic towards the fake traffic. The attacker can modify the DNS and rewrite its content so as to fool users to download unintentionally viruses to their systems.

Hussain et al. demonstrated how DNS suffers from various attacks and proposed a way to overcome these issues by implementing cryptographic asymmetric cipher algorithms to encrypt the most valuable information in communication and protect them from manipulation by attackers. Their results show better performance on preventing Denial of Service attacks (Hussain et al., 2016).

Chapter 3

Anomaly Detection Systems in ICS

3.1 Anomaly detection techniques in ICS and their current limitation

Due to the high increasing number of attacks in cyber-physical systems, several techniques have been proposed by researchers to detect and mitigate the risk of such attacks that affect Industrial control systems.

The most common approach to come across detecting and mitigating cyber attacks nowadays are the two famous Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). These systems effectively and efficiently monitor the corporate network traffic, detect any malicious activity and prevent its intrusion.

The IDS and IPS are functioning more accurately by relying on well-known attacks that have been previously stored in a traditional database. The database compares the signature of the incoming traffics and whenever convergence accrued it raises an alarm. However, the main limitation of this techniques is that it does not detect any attack which is not known by the IDS or IPS. In other words, it doesn't detect or prevent unknown attacks in the network.

In order to address the limitations in IDS and IPS, researchers have upgraded the traditional implementation to more advanced detection techniques using Machine and deep learning techniques. These algorithms support IDS and IPS and correlate the information with the corresponding application, robust the detection systems and detect anomalies with high accuracy and provide holistic analysis to the operator in charge in order to take appropriate action.

Although different Machine Learning techniques have been proposed for anomaly detection across IT and OT, the scenario in the industrial control systems possesses considerations which have to be taken into account.

3.2 IT vs OT anomaly detection

Modern ICS computing environments require different implementation of anomaly detection systems in place to ensure trusted and safe communication between IT and OT.

Host Intrusion Prevention System (HIPS) is a software package which installed on host computer and monitors a single host for suspicious activity by analyzing events occurring within that host. In other words a HIPS aims to stop malware by monitoring the behavior of code. This makes it possible to help keep your system secure without depending on a specific threat to be added to a detection update.

OT anomaly detection can fall into Network-based intrusion detection systems (NIDS) with anomaly detection capabilities on the network. Promiscuous mode on the network is mandatory in order to implement and analyze the traffic and this includes all the unicast traffic in the NIDS (Conrad, Misener, and Feldman, 2017). NIDS devices don't have the ability to interfere with the communication during malicious activities since NIDS are known to be passive devices.

3.3 Real-time vs non-real time detection

It's major importance to pay close attention to the current cyber-attacks, but more importantly is identifying the source of such an attack, bearing that in mind will allow us to know whether we should implement real-time or non-real-time monitoring mechanisms.

In order to do that, the first and foremost is to have as much information as possible about the attacks. The **non-real-time detection** technology basically are the traditional anti-analysis softwares that are functioning in an anti-statistic approach and at some point they are no longer resist the current attacks in the Industrial control systems, since the non-real-time detection applications do not instantly interact with the anomaly activities or report them in a short period of time.

Real-time detection are more advanced strategies and can resist against the recent attacks more efficiently . The dynamic anti-approach ,in the case the infrastructure requires high speed, the real-real time detection can operate and provide a robust detection mechanism.

The delay in reporting attacks and vulnerabilities among the industrial control systems can significantly cause serious damage, therefore, we need an approach that can instantly raise an alarm whenever undesirable actions are detected.

3.4 Analysis of existent anomaly detection for ICS in the Market

Products of an Anomaly detection systems provides a solution to ICS by learning the behavior and the normal activity in the network, device, application, process and the user behavior. Having learnt this information, any deviation and variation from these, the anomaly detection softwares of ICS are more likely to be identify and classify it as a cyber-attack.

Company	Country	Founded	Total Funding
SCADAfence	Israel	2014	\$10M
Nozomi Networks	United-State	2013	\$52.5M
Claratory	Israel	2014	\$100M
Kaspersky	Russia	1997	\$685M
Indegy	United-State	2014	\$36M
CyberX	Israel	2012	\$47M
ProtectWise	United-State	2013	\$24
Radiflow	Israel	2009	\$18M
BioCatch	Israel	2010	\$213.7M
NexDefense	United State	2012	\$8.1M
SecurityMatters	Netherland	2009	\$5M
RheBo	Germany	2014	€3.5M

TABLE 3.1: Comparison of existent anomaly detection for ICS in the Market (Peterson, 2017).

Chapter 4

KingFisher

4.1 Analysis of KingFisher

KingFisher (Bernieri, Conti, and Turrin, 2019) is a modern successful outlier detection mechanism that was built on Mininet using and exploiting machine learning algorithms. It's the first of its kind that can provide a solution and detect attacks in IT and OT traffic of ICS networks.

In the experiment, the authors implemented Variational Auto Encoders (VAEs), a branch of an artificial neural- network that classifies the data with no labels provided to the training data, in other words, in an unsupervised manner. The results show that KingFisher has the ability to capture attacks on network and physical layers of TCP/IP models.

Generally, the authors in the paper implemented **Denial of Service (DoS)** attack in which the attacker aims to make the system unavailable by flooding the server and consuming its resources. They also implemented the **Modbus Function Code Modification (MFCM)** attack. The idea here is that the attacker can retrieve sensitive information from the server which denotes the current state of the system. This can be done by changing the functionality of the Modbus reading and storing the registers' information. Another important attack considered in Kingfisher is the so-called **Modification of Physical Behavior (MPB)** attack. As the name implies, it modifies the physical behavior of the device and alters its functionality. Finally, the most important part of Kingfisher and the attack that has been addressed in this paper as well is the **Man In The Middle (MITM)** attack where the attacker performs ARP-poison on client side and alters every packet arrived at the server. This action is possible even without modifying the packets available locally at the host.

4.2 KingFisher Architecture

The architecture of KingFisher primarily came with four main components in its module explained as follows:

- **KF-IT** module: it's the main controller in charge of monitoring the connection between the Corporate and the Control network.
- **KF-OT**: is in charge of monitoring and controlling the Network traffic and detecting suspicious activities in the Control Network.
- **KF-PHYS**: this is the physical process that shows the current state of the system, this includes the noise, vibration and the level of the temperature as well as the power consumption in Watt meter Taking advantage of the physical side-channel data.
- **Correlation Node (CN)**: where the identification of complex attack and correlation of incoming data from detection nodes takes place.

The overall architecture and the workflow of Kingfisher can be visualized in the figure 4.1 along with the modules localization implemented in the network of ICS topology.

Giuseppe Bernieri, Mauro Conti, and Federico Turrin

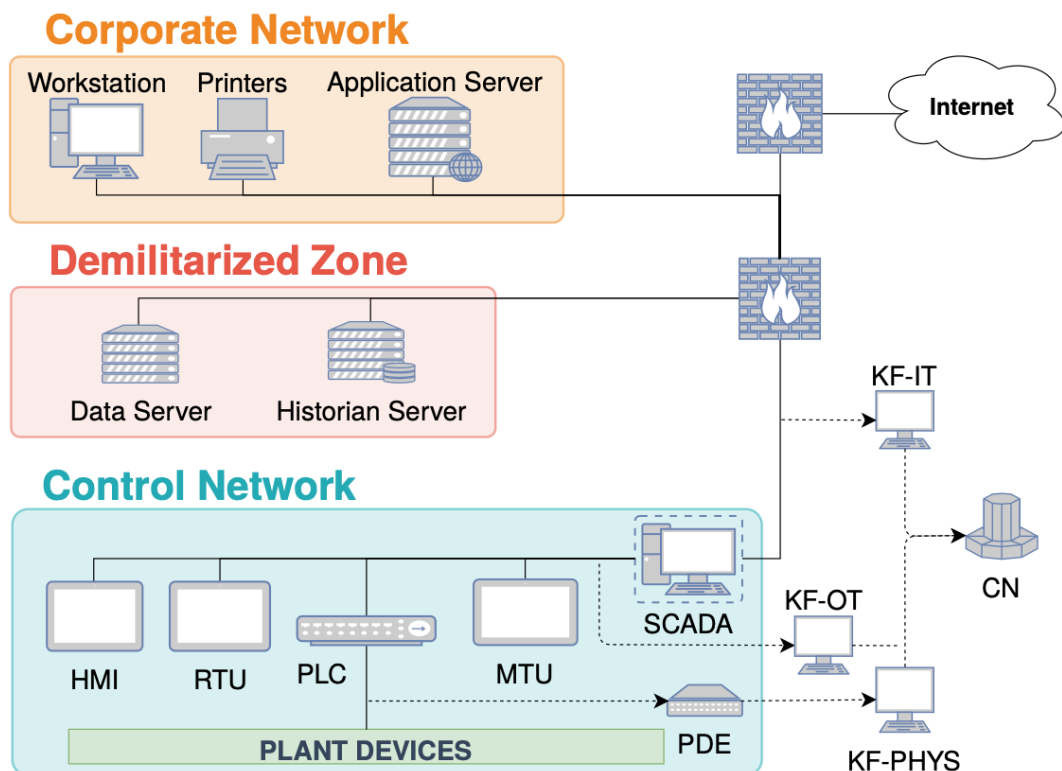


FIGURE 4.1: Workflow & architecture of KingFisher (Bernieri, Conti, and Turrin, 2019)

4.3 KingFisher Limitations

After carefully going through the kingfisher paper and understanding how the framework was built, how the architecture designed and the experiment carried out, I noticed that in the scenario, the duration in which allowed for the attack event to take place is only two minutes for the KF-OT IDS can detect anomalies and provide report and determine the communication efficiently in the model. While on the other end, the side of the KF-IT IDS, the period in this scenario is not sufficient enough to analyze, predict and provide meaningful information.

The second critical perspective suggestion in the implementation of the three Intrusion Detection Systems (KF-IT, KF-OT and KF-PHYS), due to the incompatibility of the data that are arriving to the Correlation Node (CN), leading each of the three to possess a different IDS, resulting asynchronous when collected by CN. Last but not least, the authors considered only four types of attacks which are: Man In The Middle attack (MITM), Denial of Service (DoS), Modbus Function Code Modification (MFCM) and Modification of Physical Behavior (MPB).

Finally, The application/paradigm is that it only focuses on a specific IT traffic. In other words, it doesn't identify suspicious activities in the entire system and that include application or transport layers. Also in this paper is that the authors heavily focused on modbus protocol without considering others such as DNP3.

Chapter 5

Modbus Protocol

5.1 What is Modbus

Modbus is a protocol used for transmission of information which communicates between the serial lines of electronic devices or communicating on the Ethernet. This protocol is commonly deployed in the ICS and factory automation. Modbus is an open source protocol, therefore, anyone who would like to use it on a specific network or system, they are freely able to take advantage of it at no cost. But it is worth mentioning that the Modbus has a trademark registration and it is a property of Schneider-Electric Inc. in the United State (Liu and Y. Li, 2006).. Along with the Schneider-Electric partnership, a new organization called modbus.org was established in order to further help users to use it for different purposes.

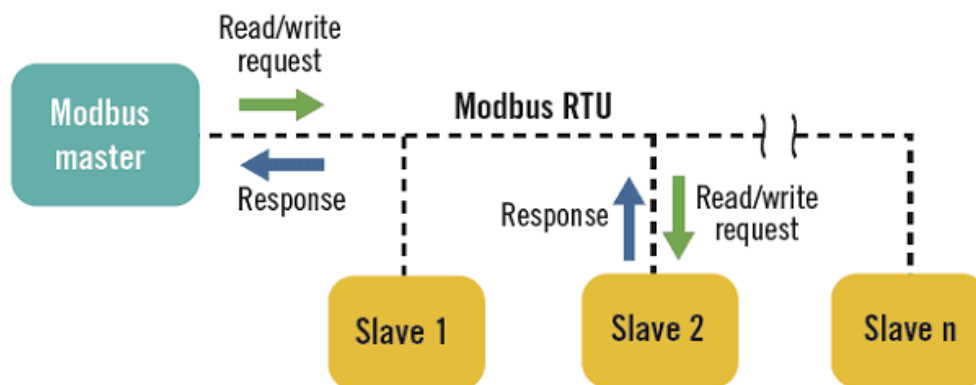


FIGURE 5.1: Description of Standard serial network of Modbus with more than 247 slaves and a single master, every slave has a unique address (William L. Mostia, 2019)

The first time when Modbus published was back in 1979 by Modicon (William L. Mostia, 2019) for the purposes of the usage of Programmable-Logic Controller. From that moment, the de-facto standard protocol for communication between the industrial electronic devices has become the Modbus protocol.

The original version of modbus is called Modbus serial protocol which usually has a slave and master. This means that for example there is a dedicated master which takes control over the transmitted data with several

slaves at the same time. The slaves on the network have to respond to the requests of the master. This request includes but not limited to, read data from the slaves or write data to slaves.

The figure 5.1 demonstrates the standard modbus serial architecture. Typically there are 247 slaves in the network and only one master that controls the communication, each and every single slave has a specific unique address. While the figure 5.2 illustrates the Modbus/TCP process.

The current version of modbus is called Modbus TCP/IP, as in the original version, the current modbus uses the term client/server interchangeably in the architecture instead of slave/master. The Modbus TCP/IP is a client and server that are communicating with each other, using the network Ethernet TCP/IP (Liu and Y. Li, 2006). The implementation of Modbus protocol is easy, however, encryption of the traffic of the protocol is a challenging task. The transaction of the data of the Modbus TCP is generally between the client and the server through an IP address. In mid 2020, The Modbus Organization replaced the term master and slave to client and server respectively.

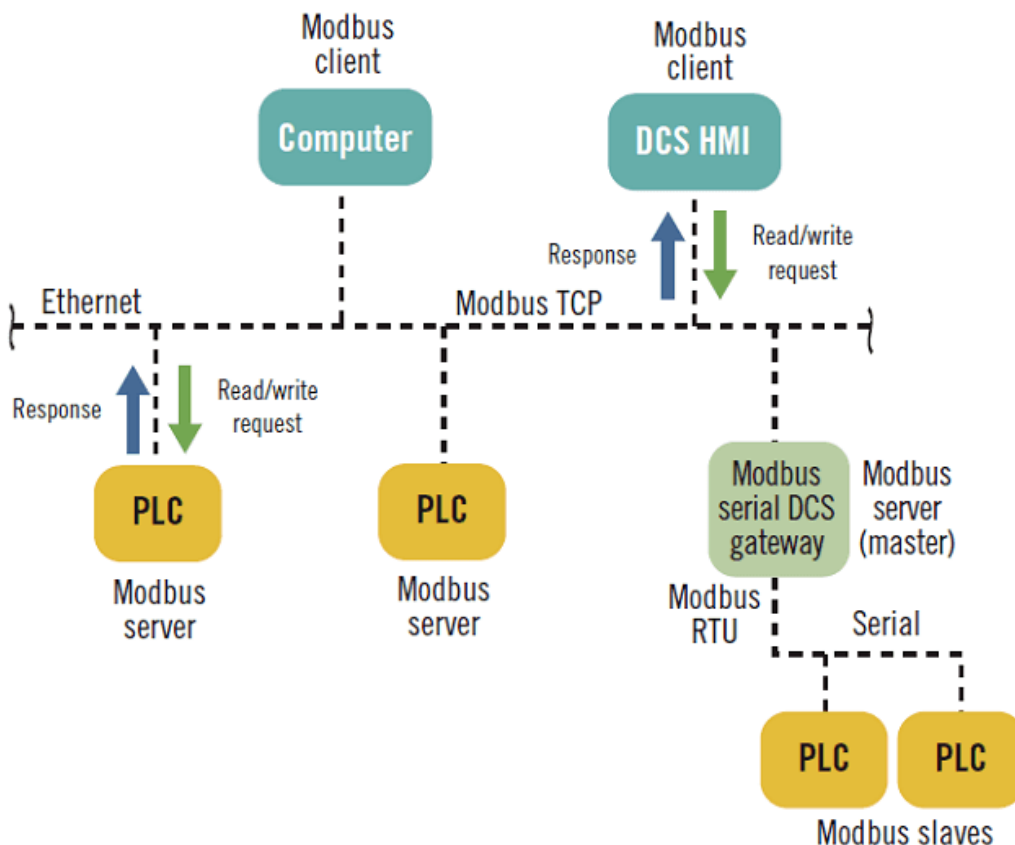


FIGURE 5.2: Description of Modbus/TCP deployed on an Ethernet Network traffic with transmitted data from server to client via IP address. (William L. Mostia, 2019)

5.2 Modbus layer

Even though Ethernet is dealing with the physical and at some point the data link layers of the ISO/OSI model (Heegard et al., 2001) in order to support the process of the Modbus, while also Modbus serial line deal with data link layer, however Modbus protocol operates at the application layer as can be seen in the figure 5.1

TABLE 5.1: Comparison between the ISO/OSI, TCP/IP and Modbus Layers (Omiccioli, 2017) .

Layer	ISO/OSI Model	TCP/IP	Modbus
7	Application	Application	Modbus Application
6	Presentation	-	-
5	Session	-	-
4	Transport	Transport	-
3	Network	Network	-
2	Data-Link	Data-link	Modbus Serial Line
1	Physical	Physical	EIT/TIA 485

As stated earlier, the modbus operates at the application layer, and it is capable of distinguishing the master/slave protocol, the below layer protocol can support to get rid of the uncertainty of the forwarded messages. The best scenario to use in the communication is the node of the master and the node of the slave. The source of the control is originally from the node of the Master which makes requests to the node of slave in the communication in sequence.

The sequence can be by first the node of the master sending a message to request formulate a communication with the node of the slave, the slave on its side, sending a message to respond to the master. The Master will send a message to request a communication after successfully the first communication has been made between the first node of the slave, this cycle of transmission will continue until the last node of the slave.

It is possible that the Master node can be integrated and connected with various resources during the communication in the center of control such as historians and databases (Huitsing et al., 2008).

5.3 Modbus actions

The various actions presented in Modbus protocol to support the communication between multiple devices can be found below as defined by the schneider electric.

- **Read-write:** to read and write the data between the client/server, this action has a "Discrete Output Coils" as an object type, in size of one bit.
- **Read-only:** to read only the data between the client/server, this action has a "Discrete Input Coils" as an object type, in size of one bit.

- **Read-only:** to read only the data between the client/server, this action has an "*Analog Input Register*" as an object type, in size of one 16 bit.
- **Read-write:** to read and write the data between the client/server, this action has an "*Analog Output Register*" as an object type, in size of 16 bit.

TABLE 5.2: Description of Stored data in Standard Modbus with corresponding actions (Modbus. Modbus Organization, 2013).

Coil Numbers	Data Addr	Type	Table Name
1-999	0000-270E	Read-Write	Discrete-Output-Coils
10001-1999	0000-270E	Read-Only	Discrete-Input-Coils
30001-39999	0000-270E	Read-Only	Analog-Input Registers
40001-49999	0000-270E	Read-Write	Analog-Output-Registers

In the table 5.2, the coil numbers do not actually appear in the communication and the messages, thus, they are considered as the names of the locations. The role of the Data addresses in the messages is distinguishing between the offset values, these values in every table do not have a similar offset. For example, in the beginning, the registered coil number is 40001 has a corresponding number of 0000 from the Data addresses.

5.4 Security issues

Incorrect date configuration and implementation in control systems of modbus such as Distributed control system (DCS) and Programmable logic controller (PLC) can lead to a potential security incident. This can potentially occur whether the Safety Instrumented System (SIS) is in place or not. Therefore, it is our responsibility to take into consideration the integrity of the data transmitted through Modbus protocol, this includes the correction of the data, error of the data and from where and towards which way the data is sent. The typical modbus has error detection mechanisms such as parity check, checksums, Cyclic Redundancy check (CRC), Longitudinal Redundancy Check (LRC) and ability to handle some errors and diagnose the entire configuration from master to client or from slave to server nodes. However, this security implementation is adequate to protect against normal attacks and basic data-transaction, and not sufficient enough to secure against dedicated cyber security incidents and other breaches occur internally. The Carrier Sense Multiple Access with Collision-Detection (CSMA/CD) is implemented in the Modbus/TCP to support the medium access control for security purpose (Liu and Y. Li, 2006)

Chapter 6

Message Queuing Telemetry-Transport (MQTT)

6.1 What is MQTT

The Message Queuing Telemetry Transport (MQTT) is a publish/subscribe, client/server messaging protocol used to transport messages through electronic devices, it is a lightweight, simple, open source and designed in an easily implementable way. These features made the usability of MQTT ideal for many applications and in different situation, this include constrained-environments for communications such as in the Internet of Things (IoT) and Machine-To-Machine (M2M) contexts where a network bandwidth is required and a small code footprint (Andy and Arlen, 2014)

The most favorable protocol to communicate with IoT and M2M is the MQTT protocol (Yassein et al., 2017). This is because it takes advantage of the publish and subscribe scheme to rig out simple configuration and flexible implementation. I have simulated the figure 6.1 which shows the publish/subscribe scheme of MQTT.

The scientist Andy Stanford-Clark from IBM along with research engineer Arlen Niper in 1999 invented the MQTT protocol. Later the MQTT in 2013 upgraded to become the standard protocol in many organizations including the Organization for the Advancement of Structured-Information Standards (OASIS).

The idea behind the publish and subscribe is the task of decoupling the messages that were originally generated by the publisher and later received by the subscriber. Both the publisher and the subscriber have the ability to operate without relying on or previously known each other (Velez et al., 2018). This is true since both the entities can run simultaneously and operate at the same time but either the entities are not pending while receiving or publishing. They have a filtering feature that distinguishes the messages in the communication and for the subscriber to get the appropriate message. Furthermore, the broker can be processed event driven and highly parallelized , this provide higher scalability.

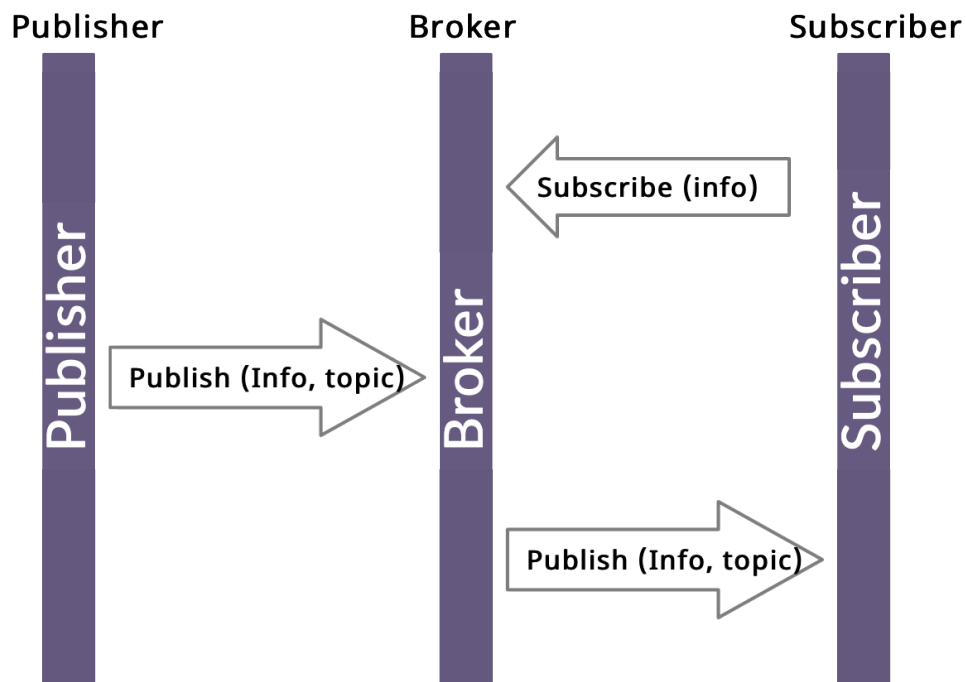


FIGURE 6.1: Process of a simple MQTT protocol utilizing a scheme of publish/subscribe

TABLE 6.1: Comparison between Brokers of MQTT protocol

Type	Address	Port	Sign-up needs
Mosquitto	test.mosquitto.org	Application	No
HiveMQ	hivemq.com	1883	No
Paho	eclipse.org/paho	1883	No
Bevywise	mqttserver.com	1883 (TCP)	Yes

Mosquitto broker is the most widely used to support the MQTT protocol, it is open source and licensed by the EPL/EDL, so far it supports the version 5.0 and 3.1. As MQTT, mosquitto is a lightweight broker and applicable to all devices that have lower single-board computing components to full servers.

Port 1883 is the default port that MQTT operates on on TCP/IP, aside mosquitto, hivemq and paho are possible brokers deployed in the configuration (Upadhyay, Borole, and Dileepan, 2016). The table 5.1 shows the famous brokers and an open source exist in the world.

6.2 MQTT Layer

Just as in the Modbus protocol, MQTT operates at the application layer of the ISO/OSI model. The figure 6.2 presents an architecture of the MQTT protocol.

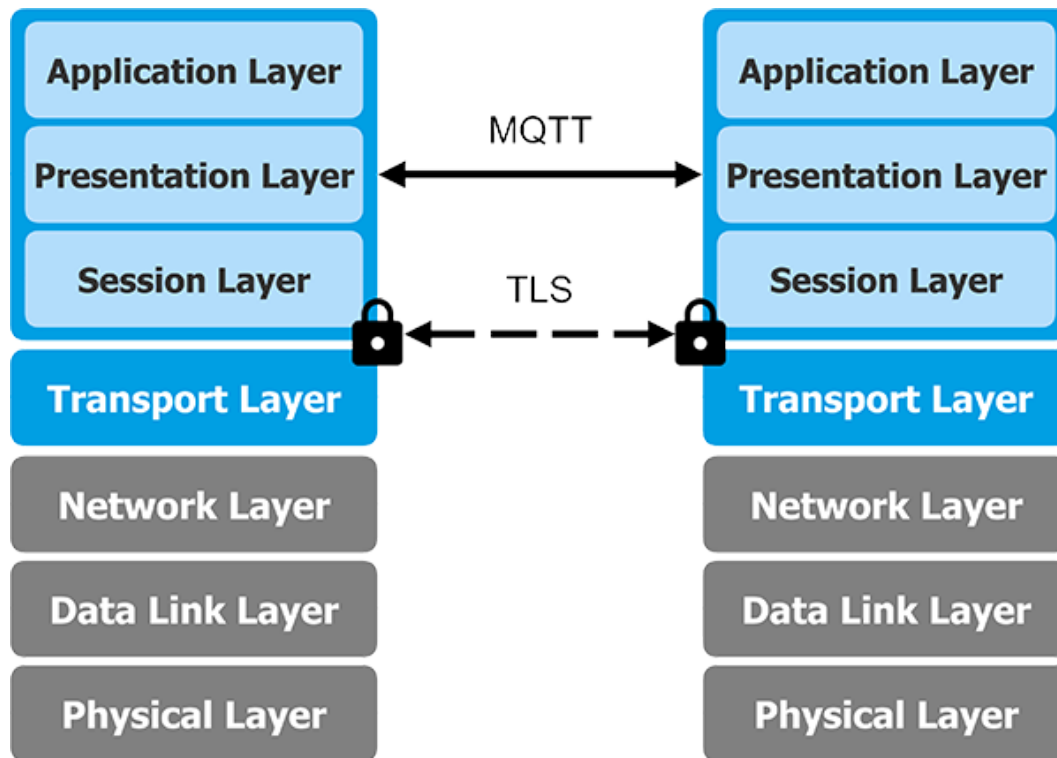


FIGURE 6.2: Description of the MQTT layer (Thiel, 2016).

If a publisher sends a message with the Retained Flag affixed, the broker temporarily saves this message for a topic. On one hand, this message is sent to subscribers who have just subscribed to the topic, on the other hand, the subscribers with number of service equal 1 who have not yet sent a confirmation of receipt will send or ignore the message. MQTT not only has a very simple basic structure, but also offers ample freedom in the subdivision of topics and does provide information regarding the content of the payload. In this way MQTT is a very generic protocol and can be used in a versatile way. There are also practically no limitations in the choice of broker: these are available and free software, from established trading companies, as a network service or hardware application. An application that uses MQTT for communication works with any MQTT broker. This ensures the ability to change platforms at any time and without particular effort.

6.3 MQTT actions

For the MQTT protocol, there are five main actions that allow users to communicate with MQTT brokers to achieve the task of interest as follows:

- **MQTT Publish:** allow users to publish a specific message.
- **MQTT Subscribe:** allow users to subscribe to a specific topic and data.
- **MQTT Receiver:** allow users to Receive a specific content of a message transmitted by the broker.
- **MQTT Unsubscribe:** allow users to unsubscribe from the topic and its data.
- **MQTT Disconnect:** allow users to disconnect from broker and terminate the connection.

These are typical actions available used to receive messages and subscribe to a topic. It is possible to extract data and information using many variables and later reusing it in a workflow.

6.4 Security Issues

The MQTT protocol is a perfect protocol, however it has some security issues, disadvantages and limitations. Among these security issues is that the MQTT protocol relies highly on the broker; the broker has many downsides in terms of the overall systems' scalability. The meaning of that is network and the architecture of the network can only be expanded as higher as the broker can handle. Not only that, the broker is also a point of failure, so if there is a chance a hacker can access the broker he can, therefore, access the sensitive data and control the entire system.

Chapter 7

Distributed Network Protocol 3 (DNP3)

7.1 What is DNP3

The Distributed Network Protocol version 3 (DNP3) as described in the DNP organization, is a public and an open source protocol. The protocol is widely used for communication in the SCADA system, remote controlling and monitoring. Since the protocol is an open source, any organization or manufacturer can configure the DNP3 protocol free of charge. DNP3 is a standard protocol for SCADA and introduced to simplify the transmissions in smart-grid nodes and substations (Amoah, Camtepe, and Foo, 2016). An important addition to this protocol has been embedded and equipped Secure Authentication security Mechanism (DNP3-SA).

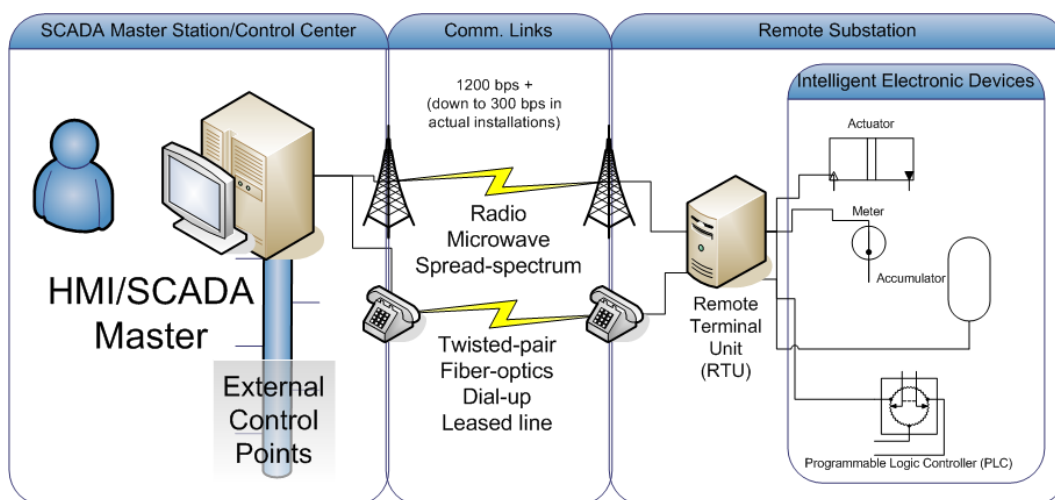


FIGURE 7.1: Components of typical SCADA system connected to DNP3 (Lemaymd, 2004)

Primarily, the communication is used between the SCADA master station, Remote terminal units or IEDs. DNP3 supports peer to peer, multiple master as well as multiple slave communications. DNP3 protocol has flexible communications and it allows users to communicate devices in many different ways.

The figure in 7.1 shows an overview of the DNP3 connected to the SCADA control center with components of an intelligent electronic devices and Human Machine Interface.

7.2 DNP3 Layer

DNP3 protocol is the most layered protocol compared to modbus and MQTT protocol. Each layer in the ISO/OSI model adds a specific functionality starting from the lowest level i.e physical layer, up to the application layer. This functionality allows the protocol to be more reliable, standardized and flexible. At each layer there are standard specifications to ensure the device is capable of implementing DNP3 protocol so as to further ensure the reliability of the connection from one device to another. The DNP organization explicitly mentioned that the DNP3 protocol was designed originally based on 3 layers of the ISO/OSI model. These layers are physical layer, data link layer and application layer as illustrated in the figure 7.2.

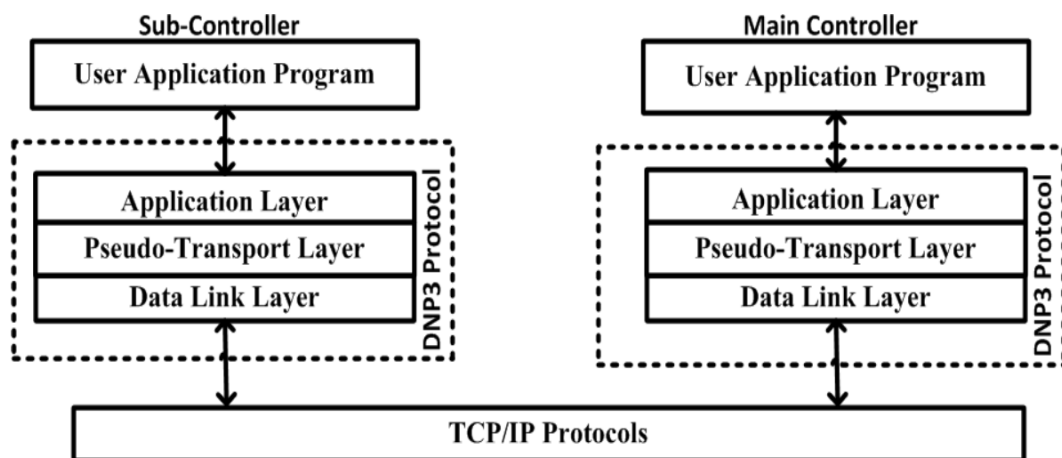


FIGURE 7.2: Description of the three DNP3 protocol and the corresponding layers of ISO/OSI model (Clarke, 2004)

Authors in (Clarke, 2004) demonstrated the DNP3 layer in detail. The transport layer has specific bytes that are added to the Transport service data-unit, after the assembling of these bytes it is later disassembled into a block of data in a fixed size. In the header, the transport layer will add one more byte and the Transport service data-unit will be formed in a segment format. Each and every Transport service data-unit will be assembled again in the Data link layer and it will become a link service data-unit and add 10 more bytes in the header. In the figure 7.3 illustrates more information about the DNP3 and related corresponding layers of ISO/OSI model.

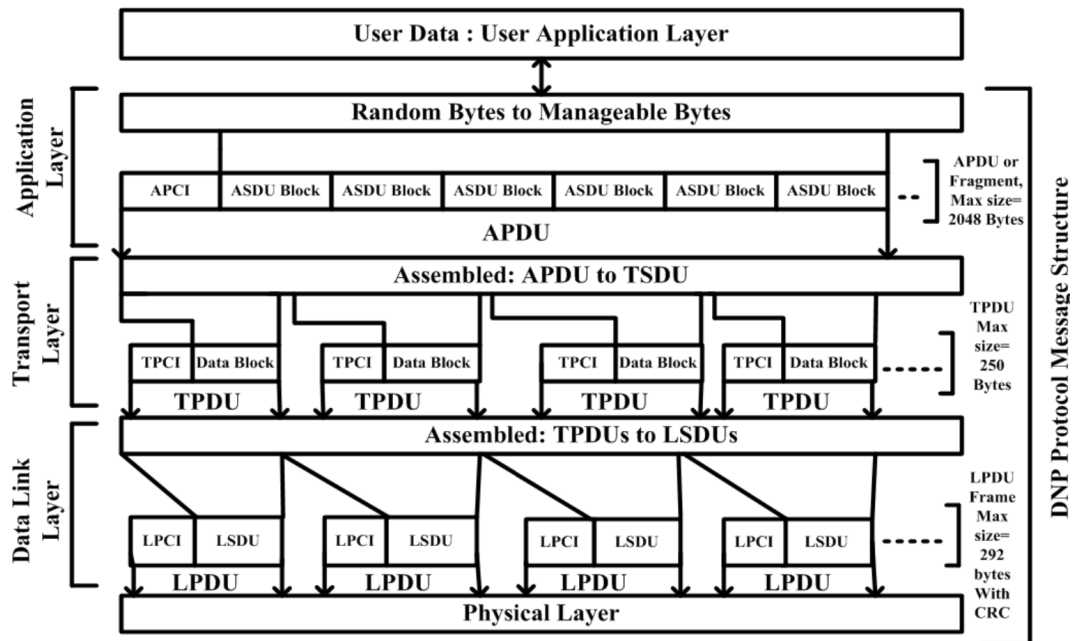


FIGURE 7.3: Description of DNP3 protocol functionalities and related corresponding layers of ISO/OSI model (Clarke, 2004)

7.3 DNP3 actions

This section contains a list of specific actions on DNP3 protocol. The protocol has more possible actions compared to Modbus and MQTT. The Schneider Electric project describe these actions as follows:

- **Initialize:** This action is associated with the counter points of the DNP3 protocol.
- **Inching:** This action is associated with the “Trip Close” of DNP3 Pulse and with “NULL” of DNP3 protocol.
- **Enable Unsolicited Events:** This action is associated with the master outstations of the DNP3 protocol.
- **Disable Unsolicited Events:** This action is associated with the counter points of the DNP3 protocol.
- **Download File:** This action is associated with the master outstations of the DNP3 protocol.
- **Delete File:** This action is associated with the master outstations of the DNP3 protocol.
- **Execute Remote Method:** This action is associated with the master outstations of the DNP3 protocol.
- **Execute Command:** This action is associated with the master outstations of the DNP3 protocol.

- **Freeze and Clear:** This action is associated with the master outstations of the DNP3 but provide support to the Frozen Counterpoints.
- **Perform Level 3 Scan:** This action is associated with the counter points of the DNP3 protocol.
- **Read Device Attributes:** This action is associated with the counter points of the DNP3 protocol.
- **Refresh:** This action is associated with the master outstations of the DNP3 and also provide status points of the DNP3 protocol.
- **Read String:** This action is associated with the master outstations of the DNP3 protocol.
- **Read Device Attributes:** This action is associated with the master outstations of the DNP3 protocol.
- **Perform Level 3 Scan:** This action is associated with the master outstations of the DNP3 protocol.
- **Reset Poll Statistics:** This action is associated with the master outstations of the DNP3 protocol.
- **Freeze:** This action is associated with the master outstations of the DNP3 but also provide support to the Frozen Counterpoints.
- **Set Update Key:** This action is associated with the master outstations of the DNP3 and also the slave outstations of the DNP3 protocol.
- **Set Clock:** This action is associated with the master outstations of the DNP3 protocol.
- **Set Value:** This action is associated with the String points of the DNP3 protocol.
- **Upload File:** This action is associated with the master outstations of the DNP3 protocol.

Other capabilities that DNP3 can provide: it can resend and request in a single message with various data types, it can also respond to unsolicited messages without a request, it allows peer to peer operations with multiple masters and it also provides support to time synchronization.

7.4 Security issues

The DNP3 protocol is very simple, reliable in terms of robustness as in the figure 7.4. However, at the time DNP3 protocol was developed, security was not a concern and a major problem. This is the reason why the DNP3 protocol doesn't have built-in security. There was no encryption or authentication for example. The lack of encryption and authentication makes eavesdropping and spoofing attacks simple, feasible and straightforward.

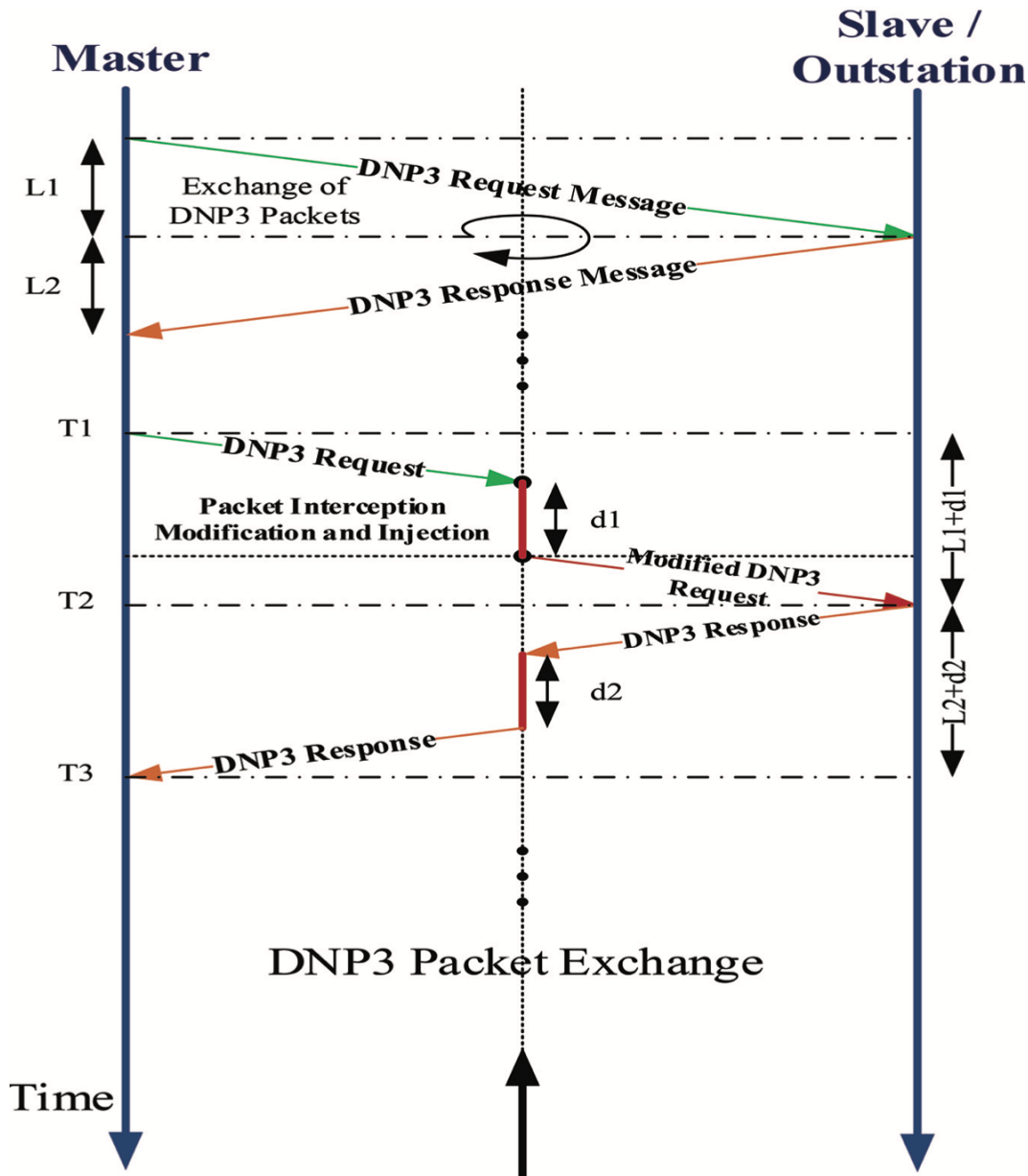


FIGURE 7.4: Simple DNP3 configuration and packet exchange (Darwish, Igbe, and Saadawi, 2016).

At the communication level, the DNP3 also is vulnerable to various attacks. Authors in (Darwish, Igbe, and Saadawi, 2016) performed several successful attacks by modifying and inject packets and manipulate the traffic,

also as in the figure 7.5, they performed Unsolicited Message Attack.

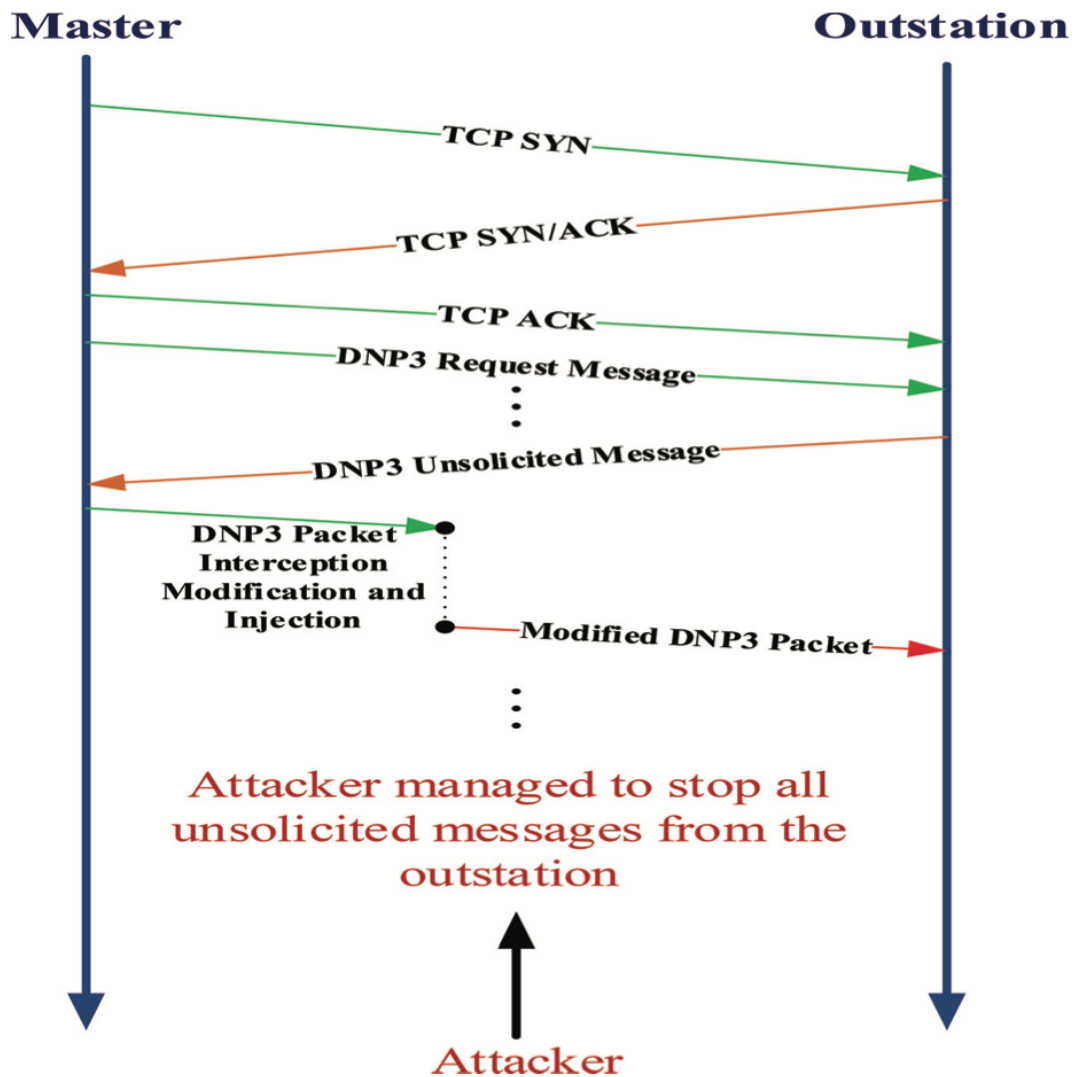


FIGURE 7.5: Unsolicited Message Attack (Darwish, Igbe, and Saadawi, 2016)

Chapter 8

Encrypted Traffic Generation

This chapter discusses the experimental set-up, mininet scenario, traffic generation and data collection with the support of wireshark analyzer.

8.1 Why Traffic Encryption is Necessary

The current ICS protocol does not include encryption or authentication mechanism, if it takes only a function code and an IP address to establish a communication, an attacker can easily capture the session and analyse the network traffic and all the exchanged requests made between the client server, master slave or master outstation as the communication is not encrypted both entities can be impersonated. Furthermore, he can access sensitive information and figure out the name of the protocol and the type of actions performed in the network. This could also be possible since Modbus, MQTT and DNP3 are open sources and publicly available online.

There are also a huge number of vulnerabilities and threats that need to be taken into account that solutions should be provided and or mitigated. Even if the data is at rest in a client or server, attackers can launch an attack and compromise the system, these attacks include rerouting the traffic, Denial of Service(DoS), Modification and Injection and Man in the Middle (MITM) attack.

8.2 Experimental set up

The experiment of this thesis was done on an Ubuntu system 20.04.2.0 LTS running as a virtual machine, the host system is a macOS Big Sur version 11.2.3 with 8 GB of memory and Dual-Core Intel Core i5 processor. It is worth mentioning that part of this experiment was performed on the server of the university of padua.

8.3 Mininet

Mininet (Keti and Askar, 2015), is a public and an open source network emulator that allows researchers to conduct experiments, debug and test networks and allows them to create their own convenient environment for simulating real network behavior. The simulation includes creating virtual switches, hosts, links and controllers. The switches presented in mininet are compatible with OpenFlow which added flexibility to customize the communication as well as Software Defined networking. Implementation of the mininet, the way it works and more details could be found on the mininet original website.

8.4 Mininet Scenario

In this experiment, Modbus protocol created with a client resides on h1 with an IP address 10.0.0.1, while the server on h2 with the corresponding IP address 10.0.0.2, and it uses the default port 502 of the TCP/IP for both parties. This architecture is exactly the same as the DNP3 protocol but with Master and Outstation and using 8080 as a port number. The MQTT protocol however, has a different scenario since it has an additional third party involved in the communication which is the broker. The client also called publisher has an IP address 10.0.10.1 resides on h1, the server has an IP address of 10.0.20.1 and resides on h2, while mosquitto runs as the broker with an IP address 10.0.10.10. The diagram presented in the figure 8.1 demonstrates the entire simulation of the mininet architecture.

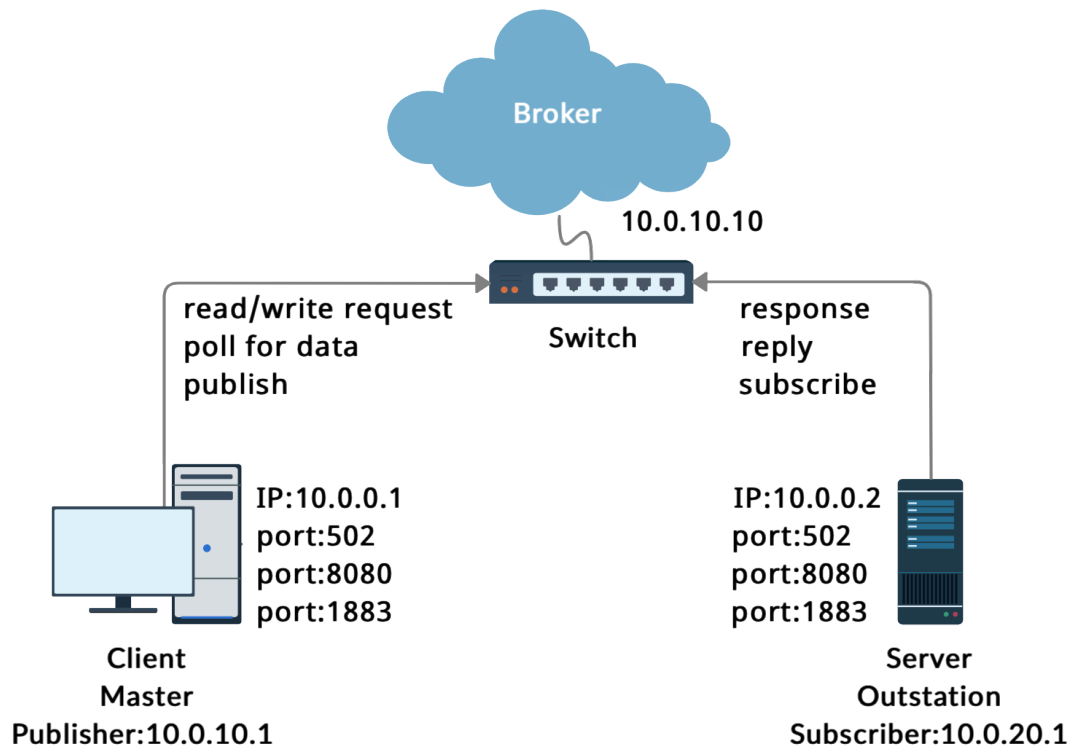


FIGURE 8.1: Graphical representation of the mininet scenario and the architecture of the simulation.

8.5 CICFlowMeter

One of the famous network traffic analyzers nowadays is CICFlowMeter. It generates more than 84 traffic flow network features, and can read any files in a pcap format and can graphically provide a report of the extracted features and convert it into a CSV file. CICFlowMeter was distributed by Canadian Institute for Cyber security (CIC) as an open source program and programmed using Java programming language and it is freely accessible on GitHub repository (Habibi Lashkari, 2018). It can be deployed to generate flows in bidirectional format, where the source to destination is the first direction which determines the forwarding packet, while the destination to source is the next direction and determines the backwarding packet on the network, and upon the connection teardown, the TCP flows are terminated. CICFlowMeter provides flexibility to calculate the desirable features and remove or add additional ones, it also offers integrity and allows you to better control the timeout and the duration of the flow.

8.6 Traffic Generation

After implementing the ICS protocols, in my analysis, it is possible to read the entire communication, therefore, an attacker can notice the transfer in clear text of all the messages over the media transmission.

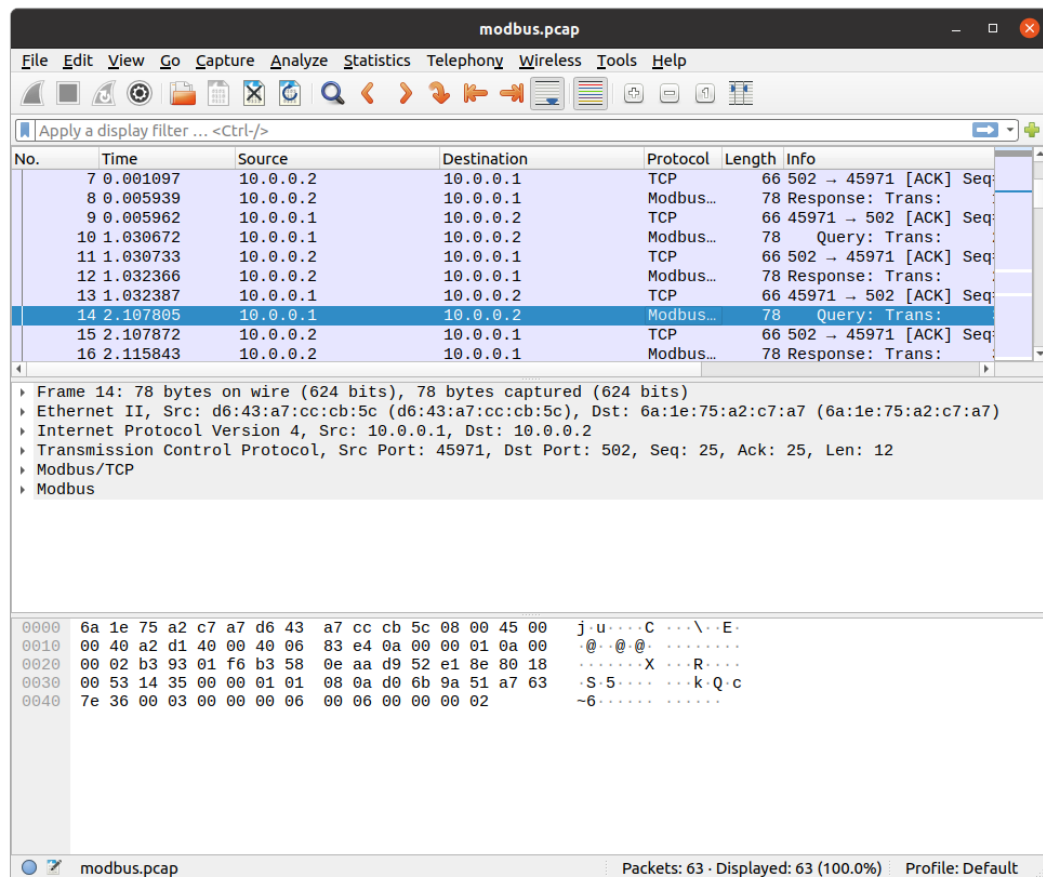


FIGURE 8.2: Unencrypted Modbus Traffic

It can be noticed below the figure 8.2 in a wireshark format that contains read coils and the requests of a Modbus protocol presented in clear text. This denotes that in Modbus protocol there is a lack of security and confidentiality. Not only that, but also there is no integrity check available and no authentication involved. An attacker can easily perform various attacks on this network traffic, the attack could be a sniffing on the traffic and identify the protocol and all the sensitive information about the devices that are using this protocol on the network. Since there is no encryption in place, the attack of issuing a harmful command is feasible.

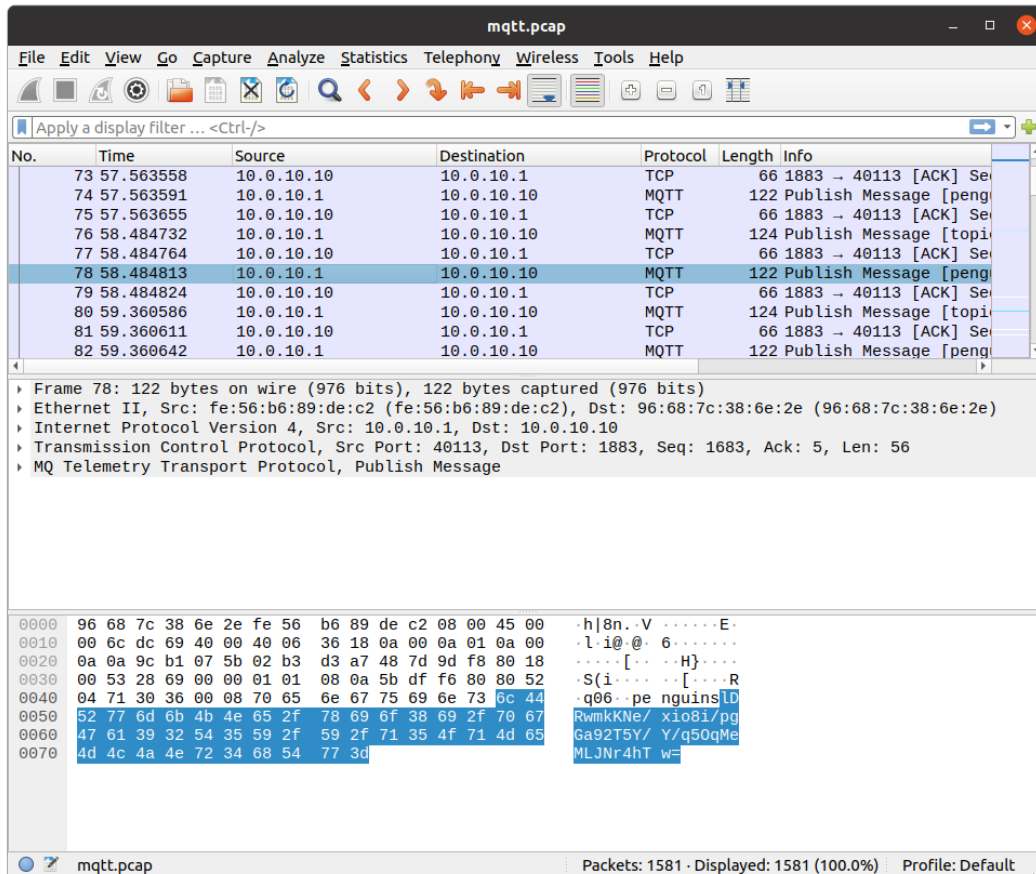


FIGURE 8.3: Unencrypted MQTT Traffic

The lack of security, confidentiality and integrity associated with Modbus protocol, the same issues also applies to MQTT and DNP3 protocols. The figure 8.3 shows MQTT protocol without encryption in place. We can vividly see the name of the protocols along with sensitive information transmitted in the network.

The screenshot shows the Wireshark interface for a capture file named 'dnp3.pcap'. The main pane displays a list of 10 packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	10.0.0.1	DNP 3.0	83	Unsolicited Response
2	0.001705	10.0.0.1	10.0.0.2	DNP 3.0	90	Disable Spontaneous M
3	0.002143	10.0.0.2	10.0.0.1	DNP 3.0	83	Response
4	0.024662	10.0.0.1	10.0.0.2	DNP 3.0	81	Confirm
5	0.032073	10.0.0.1	10.0.0.2	DNP 3.0	87	Write, Internal Indic
6	0.032529	10.0.0.2	10.0.0.1	DNP 3.0	83	Response
7	0.053914	10.0.0.1	10.0.0.2	DNP 3.0	93	Read, Class 0123
8	0.054583	10.0.0.2	10.0.0.1	DNP 3.0	358	from 10 to 1, len=255
9	0.057669	10.0.0.2	10.0.0.1	DNP 3.0	230	Response[Malformed Pa
10	0.089143	10.0.0.1	10.0.0.2	DNP 3.0	90	Enable Spontaneous Me

The detailed view for the first packet (No. 1) shows the following structure:

- Frame 1: 83 bytes on wire (664 bits), 83 bytes captured (664 bits)
- Ethernet II, Src: 42:db:d9:e2:54:0c (42:db:d9:e2:54:0c), Dst: 1e:bf:d8:d5:4f:9b (1e:bf:d8:d5:4f:9b)
- Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1
- Transmission Control Protocol, Src Port: 20000, Dst Port: 34819, Seq: 1, Ack: 1, Len: 17
- Distributed Network Protocol 3.0

The hex dump shows the raw data of the packet:

```

0000  1e bf d8 d5 4f 9b 42 db d9 e2 54 0c 08 00 45 00  ...0 B...T...E
0010  00 45 17 3a 40 00 40 06 0f 77 0a 00 00 02 0a 00  .E.:@.@...w.....
0020  00 01 4e 20 88 03 f8 c2 57 cf d6 ec e3 0f 80 18  ..N....W.....
0030  00 55 14 3a 00 00 01 01 08 0a 6a 1d 89 04 c4 c2  .U.:...j.....
0040  31 fb 05 64 0a 44 01 00 0a 00 6e 25 c0 f0 82 80  1..d.D...n%....
0050  00 6b 7d                                     .k}

```

The status bar at the bottom indicates: Packets: 16 - Displayed: 16 (100.0%) Profile: Default

FIGURE 8.4: Unencrypted DNP3 Traffic

From the figure above in 8.4, it is clear to visualize the name of the protocol which is the DNP3 protocol, the length of the packet and more importantly the type of the messages and actions that have taken place.

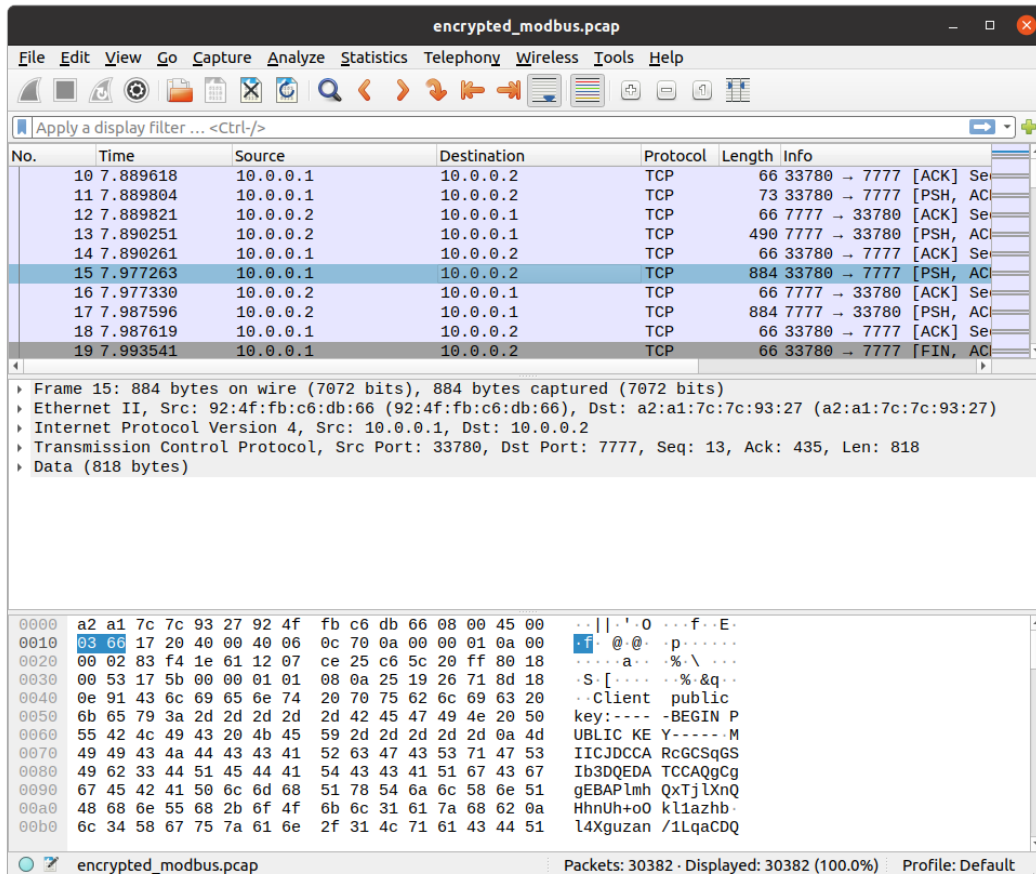


FIGURE 8.5: Encrypted Modbus Traffic

Taking the advantage of the Advanced encryption standard (AES) and Ron Rivest Adi Shamir Leonard Adleman (RSA) cryptographic algorithms, effectively able to perform encryption on Modbus, MQTT and DNP3 protocols.

After successfully implementing encryption on the network traffic, the protocol now in secure and all the contents of the message cannot be compromised, the data and information in the packets are in an encrypted format, therefore, the attacker cannot read traffic From the previous figure, the pcap wireshark file shows insecurity of the traffic, while in the figure 8.5, the traffic is completely secured.

8.7 Data Collection

The data was collected on a Virtual Machine for three different ICS protocols, the protocols are Modbus, MQTT and DNP3 protocols. It takes a matter of seconds to generate one flow with a specific action, in order to generate a sufficient amount of flows, several commands have to be launched, this is exactly the case at the time of collecting these datasets.

Generally, after collecting the datasets, wireshark analyzer presents noise in the data and undesired packets such as arp request and arp reply, this may badly influence the accuracy of the results when modeling the data. As a result, the data was properly filtered from the noises and removed all the undesired packets presented during collection and manually labeled and saved the data properly and ready for the next phase.

At this stage, the format of the data collected is in a .PCAP file, it is possible to proceed and implement some machine learning techniques and acquire the desired results, however, based on my experience, dealing with a .CSV file is much more efficient than the PCAP file. Therefore, CICFlowmeter (Habibi Lashkari, 2018) was used to perform an offline analysis and used to convert the file into CSV format.

Chapter 9

Machine Learning Techniques

Machine learning technologies nowadays are the most trending techniques used in the world of data analytics, Human interaction and artificial intelligence.

9.1 Supervised & Unsupervised Learning

In general, there are two approaches to machine learning algorithms, the supervised and the unsupervised machine learning algorithms. In the supervised algorithms, it is necessary for the user to label the data, classify the data and pass them into the algorithm, therefore, the supervised algorithm will learn from the labeled data in order to make the intended classification or regression.

On the other hand, unsupervised machine learning is unlike supervised learning. Here the intervention from the user is not necessary, therefore, the data is entered without prior knowledge or labeling. The unsupervised algorithms are divided into two main categories, the first and the one I used in this thesis is clustering which grouped the output of the dataset as I will describe in the next section of this chapter.

The figure 9.1 demonstrates the machine learning techniques along with possible algorithms used for modeling and simulation.

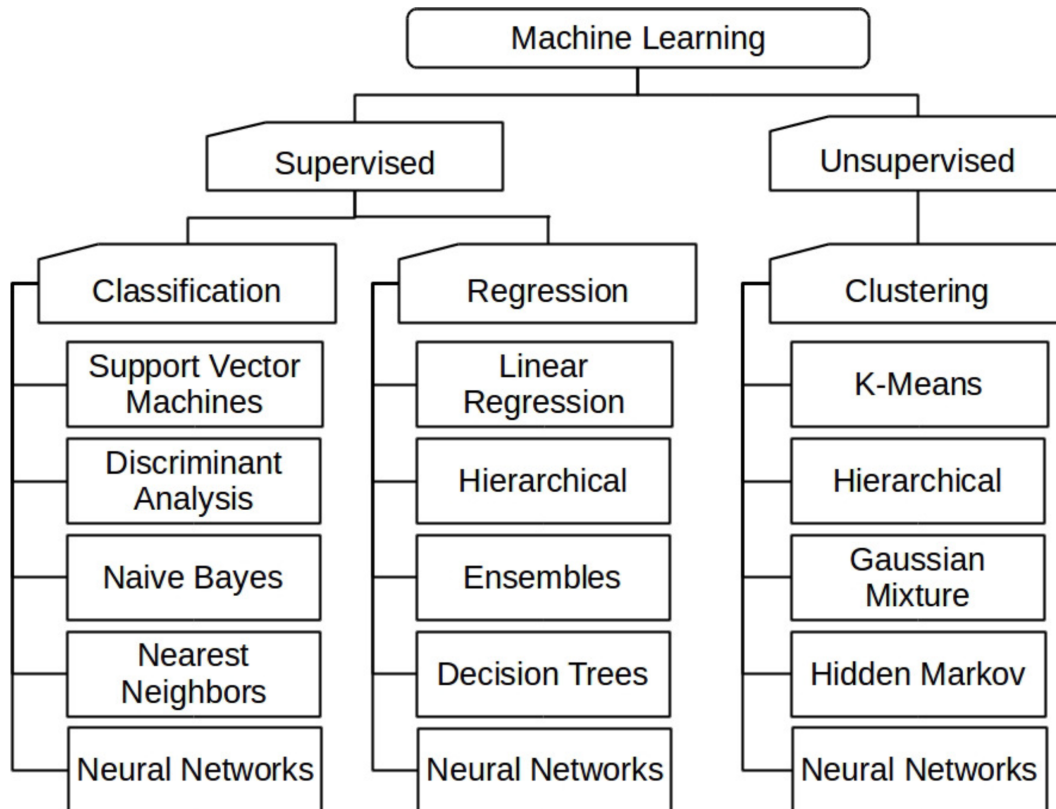


FIGURE 9.1: Categories of the most commonly used Machine learning algorithms (Duc et al., 2019).

Both the supervised technique and unsupervised technique have been used in this thesis. For unsupervised learning, I have used Hierarchical Agglomerative Clustering in order to preprocess the generated dataset. The idea behind this procedure is that each observation or flow will create a cluster that belongs to itself. For every iteration, the clusters are paired and merged to the nearest cluster and move to the next, this operation of combination is governed by a distance metric between the flows of the pairs using the DTW, the operation will continue until the entire number of clusters is iterated (200 clusters in my case).

The output from the dataset is a list of numeric vectors in .txt files and an additional file containing only the actions in .txt files.

The classification part utilizes supervised learning within its procedures. After the process of clustering the vectors and actions of the protocols in the dataset, the supervised learning will take place in order to train and test the dataset, taking advantage of the Random Forest, Support Vector Machine and Neural Network algorithms.

9.2 Unsupervised Learning

9.2.1 Clustering

In the typical clustering scenario, the aim of the algorithm is to find k groups that are called clusters, where the inter-cluster similarity is very low while the intra-cluster similarity is maximized, meaning that samples belonging to one clusters are very similar to each other while on the other end, the clusters still being very different from samples belonging to other clusters.

Take into account that the number of k clusters and the similarity function between samples have to be explicitly defined by the teacher and both have a significant impact on the final result. The input of the clustering algorithm is usually the entire dataset while the result is a list of integers where N is the total number of samples and $C \in [1; k]$ is a an integer number that denotes the cluster to which the i - th sample has been assigned by the algorithm.

9.2.2 Hierarchical Agglomerative Clustering (HAC)

In my implementation, I have used the Hierarchical Agglomerative Clustering (HAC). Generally the HAC algorithm creates clusters in the dataset for each flow by reducing the number of clusters and merging the nearest clusters at each step together with the next flow based on the flow duration and the distance function. Whenever the number of clusters of K value is reached, the algorithm will stop at that point.

Dendrogram: tree, with input points $\mathbf{x} \in \mathcal{X}$ as leaves, that shows the arrangement/relation between clusters.

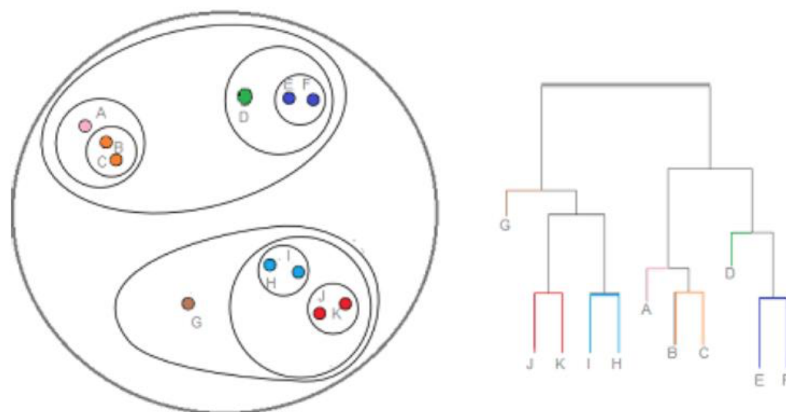


FIGURE 9.2: Hierarchical Agglomerative Clustering Structure (Zanuttigh, 2020)

In order to better understand the structure of the HAC algorithm, the figure in 9.2 exploits the functionality of the HAC, usually the output is referred to as a dendrogram or nested cluster. The number of k clusters used in my experiment is set to 200, and *fastdtw* is used to compute the similarity of the distances between the instances.

In general, the function which addresses clusters distance computation can be categorized as follows;

- Single linkage based clustering.
- Max linkage based clustering.
- Average linkage based clustering.

Take into account that neither the distance metric of the instances nor the linkage criteria are fixed, therefore, I have used the average linkage based clustering of the DTW given two clusters u and v and the metric DTW, $d()$ is computed with the following equation;

$$d(u, v) = \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} d \frac{(u[i], v[j])}{|u| * |v|}.$$

Where $d()$ refers to the distance function and the u, v values refers to the clusters of the n, m instance values respectively.

In order to compute this equation, Scipy provides this package of clustering, as "*scipy.cluster.hierarchy*" with time complexity of $O(N^2)$ for each observation. To describe the merging procedure, firstly all the distances between instances must be computed, then items are less and close to each other are merged in a single pair cluster, at the end, the distances from the cluster to all the other points are computed until the end, note that some of the instances are also further clustered in subsequent iterations.

9.3 Supervised Learning

9.3.1 Classification

After finalizing the unsupervised and clustering phase, and obtaining the suitable dataset, now it is possible to apply the supervised learning techniques. The typical way to apply the supervised learning is to divide the dataset into two main categories after labeling the data correctly as follows;

- **Training set;** use to pick an hypothesis.
- **Testing set;** it is used to test the algorithm and measure its accuracy. Sometime the data is further derived into
- **Validation set;** use to estimate the true error of the hypothesis.

It is worth mentioning that all the three sets above must be independently distributed according to the desired classification percentage (usually 70%, 15% and 15% for training, validating and testing set respectively), otherwise if an observation overlapped in another set then the entire performance of the model will display biased prediction.

9.3.2 Random Forest

The idea behind Random Forest Algorithm is that it takes a small amount of weak classifiers, combines them together and generates stronger classifiers. Random Forest (also known as Decision Trees) in its model, takes a random portion of the guess output and the test sample, later the model will compute the average of all the sample and the guess and finally produce the final decision.

The process of Random Forest Algorithm can be visualized in the figure 9.3 To derive the functionality of Random Forest, Scikit learn provides `sklearn.ensemble.RandomForestClassifier` package, which I have implemented in my experiment. It provides flexibility to customize the decision trees parameters in the model.

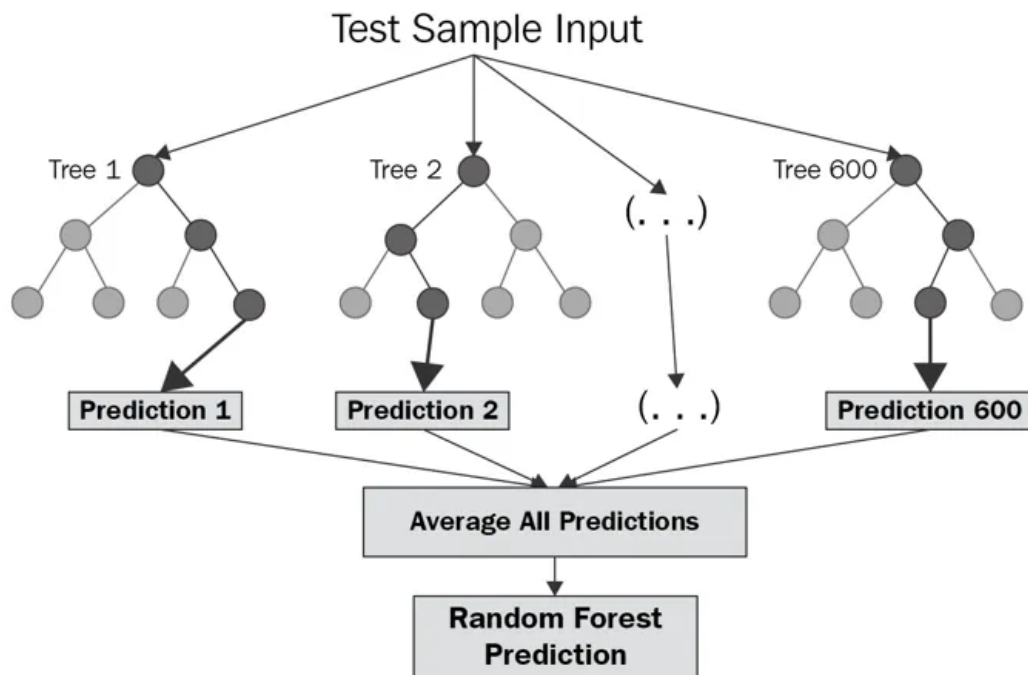


FIGURE 9.3: Random Forest Classifier's workflow, utilizing 600 weak trees learners (Chakure, 2019).

9.3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning model, the framework of the algorithm analyzes the data to apply classification or regression. SVM assumes that there is a linearly separable data (i.e. there exists a halfspace that can classify the training set perfectly) and finds the best separating hyperplane among others.

In this case we can apply the *Hard-SVM* which seeks for the largest margin when separating the hyperplane with computational problem as follows;

$$\arg \max_{(w,b: ||w||=1)} \min_i y_i (< w, x_i > + b).$$

Where a linearly separable training set $S = ((x_1, y_1), \dots, (x_m, y_m))$ exists if a half space (w, b) . such that $y_i = \text{sign} < w, x_i > \forall_i = 1, \dots, m$.

The minimization problem of hard-SVM

$$w = \arg \max_w ||w||^2 \text{ subject to } \forall_i : y_i < w, x_i > \geq 1.$$

Can be rewritten as a maximization problem

$$\max_{\alpha \in \mathbb{R}^m: \alpha \geq 0} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j < x_i, x_j > .$$

The formula is called “*dual*” problem. It’s very important for the “*kernel trick*”. The idea behind it is that it does not require direct access to the instances, instead, only the inner product between the instances.

The limitation of Hard-SVM is that it is necessary to apply this method only on linearly separable data, this assumption almost never true in real world problems.

To overcome this limitation, we need an approach that can be flexible and applicable on non linearly separable data.

Soft-SVM is one solution to this problem, it relaxes the constraints of Hard-SVM but takes into account the violations of the separation into the objective function.

It introduce the *Slack Variable* $\zeta = (\zeta_1, \dots, \zeta_m)$,

where $\zeta \geq 0$ and $\forall i = 1, \dots, m : y_i (< w, x_i > + b) \geq 1 - \zeta_i$

and the ζ_i demonstrate how much the constrain is violated. in the hyper-plane.

The input of the Soft-SVM optimization problem is $(x_1, y_1), \dots, (x_m, y_m)$, with the parameter $\lambda > 0$ which solve;

$$\min_{w, b, \zeta} \left(\lambda ||w||^2 + \frac{1}{m} \sum_{i=1}^m \zeta_i \right).$$

and subject to $y_i (< w, x_i > + b) \geq 1 - \zeta_i$ where $\zeta_i \geq 0$.

The Soft-SVM jointly minimize the norm of w (which results in maximizing the margin), and also minimize the average of ζ_i (which results in minimizing the constraint violation) Finally, the trade off between these two minimization objectives is controlled by $\lambda > 0$.

More detail on SVM can be found in the book titled “*Understanding Machine Learning: From Theory to Algorithms*” (Shalev-Shwartz and Ben-David, 2014).

9.3.4 Deep Neural Network

The Deep Neural Network classifier is a computational model inspired by the structure of Neural Network in the brain, it is based on a larger of multiple layers (called perceptron or neurons) connected to each other and takes a feature of vector as an input returns a binary output.

The DNN is generally represented in a form of directed graphs, where the edges linking between the neurons and the nodes correspond to the neurons themselves.

The simplest neural network is the *Feedforward Neural Network* in which its network is represented as a graph with no cycles, this means that the information only flows in direction.

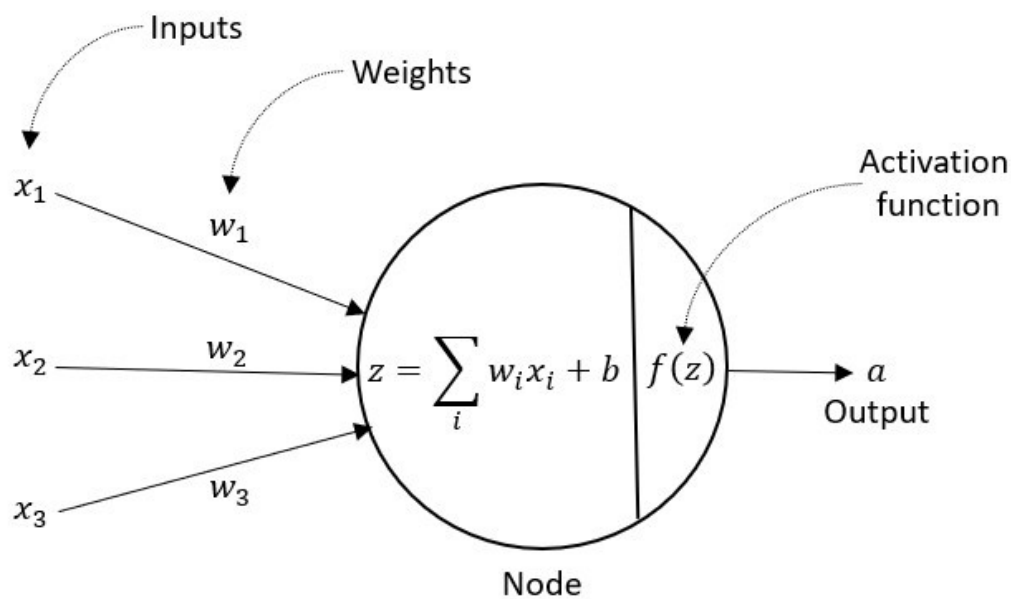


FIGURE 9.4: Simplified feedforward Neural Network scheme (Larson, 2020).

The draw in figure 9.4 demonstrates one single node that take an input as the sum from previous layers of the connected neurons and weighted by the edge weight (w), or take an input from the next layer called activation function.

There are various activation functions that can be exploited for neural networks and Convolutional Neural Network (CNN) as the following;

- Sign & Threshold Function: $\sigma(a) = \text{sign}(a)$.
- Sigmoid Function: $\sigma(a) = \frac{1}{1 + e^{-a}}$.
- Hyperbolic Tangent Function: $\sigma(a) = \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} = \frac{e^{2a} - 1}{e^{2a} + 1}$.
- Rectified Linear Unit(ReLu) Function: $\sigma(a) = \max(0, a) = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$.

Chapter 10

Data Processing & Modeling

This chapter discusses the structure of the collected dataset, data preprocessing, clustering, model classification, Dynamic Time warping, classification metrics, learning framework and a few supervised machine learning algorithms.

10.1 Dataset

The dataset used in this experiment is the result of the generated and collected network traffic from the previous chapter, with a pool of user actions belonging to different ICS protocols. The dataset were presented in a pcap file but later converted to .CSV file and it contains various traffic flow with multiple rows with time ordered sequence of TCP packets and belong to a specific user action and each action may have one or more flows, and some instances of the same action may be composed of different number of flows. The most important and useful field presented in the dataset are as the following items:

- **Protocol Name:** it shows the name of the protocol in the flow.
- **Label:** it shows the type of user actions of the protocol.
- **Flow Duration:** it contains the duration of the flow.
- **Flow ID:** the unique ID address of the flow.
- **Src IP:** provide information about the IP address of the source.
- **Src Port:** provide information about the port address of the source.
- **Dst IP:** provide information about the IP address of destination.
- **Dst Port:** provide information about the port address of the destination.

The fields used in the experiment are the Flow Duration, Protocol Name and the Label field.

10.2 Data Preprocessing

The main goal of the data preprocessing phase is the conversion of the user actions presented in the dataset to instances that can be manipulated and further processed by the machine learning algorithms. The so called instances goes by the specific names of feature vectors, these vectors are in in a shape of *n-dimensional vectors*, that's to says: $X^T = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ and the corresponding domain space is called *descriptor space*, each and every valued entered it stores important data that are useful for classification by machine learning. The process of the enter procedure is categorized as follows:

- **Flow Dataset processing:** this reads the entire dataset and creates new flow data which only concern data that belong to a single protocol.
- **Clustering:** for each cluster, the leaders are computed with the flows and are associated with the previous data.
- **Feature Vectors:** it computes the output of a feature vector for every action.
- **User Actions preprocessing:** it finds the actions in the data and computes it along with the vectors in the dataset.

The fields used in the experiment are the Flow Duration, Protocol Name and the Label field.

The purpose of the Flow Dataset processing is to enable the data to produce flows which are similar to the initial dataset, but will make the task easier to handle the clustering correctly. First of all, the flows are taken from the specified protocol and only the packet in the *Flow Duration* column is considered for this task. Later, the flow processing will be computed and extract the relevant intervals that are most likely to hold essential information for clustering.

The next step is the clustering phase, which is computed by using Hierarchical Agglomerative Clustering, taking the advantage of the Dynamic Time Warping (DTW) metric and using the Average linkage based clustering. The DTW metric is also used to represent the flows for each cluster and find the instances that minimize the overall distances between one cluster to another with the following equation.

$$\arg \min_{f_i \in C} \left(\sum_{j=1}^n \text{dist}(f_i, f_j) \right).$$

Notice that all the actions have been correctly verified by creating a list in which every entry of the list is an action flowesly isolated in for loop iteration, while all the non important or not considered actions are labeled as *other*.

10.3 Dynamic Time Warping (DTW)

The Dynamic Time warping denoted as (DTW) is method used to differentiate the temporal sequence between two time series and find the optimal similarity in terms of speed, distance and length.

A quite simple example could be reading a paper by two students and allowing the algorithm to find out who read faster than the other. The DTW has been deployed in many different applications such as speech recognition and signature recognition.

In this experiment, DTW has been deployed to measure the similarity between two flows as a set of time series.

The **Time Series** metric was deployed in the *Flow Duration* filed, and since this packet is a sequence of integers, the flows is set to count the duration and the distance between them using the DTW given two different type of series as follows:

$$X = (x_1, x_2, \dots, x_n), Y = (y_1, y_2, \dots, y_n).$$

In order to compute the cost of the Matrix, the value denoted as $C \in \mathbb{R}^{N \times M}$, where the entry of each distance C_j, k refers to the duration and the distance between the x_j and y_k

In addition, a warping path is a sequence of entries that link $[1, 1]$ to $[N, M]$, by creating a monotonic path of adjacent entries. The cost of the path is calculated by summing all the entries in C belonging to the corresponding path. Furthermore, the warping path which costs less is called the optimal warping path and it is the measure used to evaluate distances between time series which is computed as $DTW(X, Y)$. In my thesis, I have computed the optimal path of the DTW by taking the advantage of the new algorithm called **fastdtw** python library that acts and provide the optimal solutions same as DTW but with time complexity of $O(N)$.

10.4 Classification Metrics

After the clustering phase, the classification phase takes place. In order to optimize the model in classification, measuring how effectively the classifiers can correctly predict the expected result is a crucial part. The following difinition are usually used to refer to the classification metrics in order to analyze the classifiers:

- **True Positive:** when both the observation and the prediction is positive.
- **False Negative:** when both observation and the prediction is negative
- **True Negative:** when the observation is positive but the prediction is negative.
- **False Positive:** when the observation is negative but the prediction is positive.

$$\mathbf{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}.$$

$$\mathbf{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}.$$

$$\mathbf{F1\ Score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}.$$

$$\mathbf{Accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}}.$$

Bear in mind that the above classification metrics are applicable on both specific classes as well as the entire model, which correspond to the computation average of each class.

Finally, Sensitivity Matrix (also known as Confusion Matrix) can be used to measure these metrics and visualize the evaluation of the performance of each class in the model. Scikit learn (Pedregosa et al., 2011) provides the machinery to easily calculate the classification metrics effectively.

10.5 Training & Testing modeling phase

Having done the classification metric, the next phase is to train and test the dataset using the machine learning algorithms, the aim of this procedure is to take the outcome of the clustered data (feature vectors and user actions) and apply the ML techniques. Since user actions in the clustered data is a multiclass classification problem, I will perform the supervised learning technique, taking advantage of Random Forest, Support Vector Machine (SVM) and neural network.

The classification procedure implemented in this thesis are as follows:

- Dataset division into two sets , training and test set and computing the feature vector into a subset of 80% for training the model and the remaining 20% for testing.
- Building the classifier and selecting the best parameters in order to choose the best features according to the testing score.
- Applying the classification metrics and compute the precisions, recall and F1 score.
- Applying Confusion matrix and visualize the model made a wrong prediction in the model.

Chapter 11

Results & Analysis

The experiment and the entire framework implemented using python programming language; after data collection phase, the experiment contains two main codes 1) clustering.py which contains the first part of preprocessing the dataset, 2) classifier.py which contains the machine learning algorithm, testing and training the model.

For each protocol, the analysis is done separately and it takes into consideration a pool of several actions for three different ICS protocols which are Modbus, MQTT and DNP3 protocols.

The following table 11.1 contains user actions taken into account for each protocol.

Notice that this is a multi class classification problem, and in multi class classification, the Machine learning take instances of one, two, three more classes and transforms it into binary classification.

Also notice that the “*other*” action is a common action that’s labeled for all the other protocol’s actions that were not taken into consideration.

TABLE 11.1: User actions taken into account for each protocol

Modbus	MQTT	DNP3
modbus-read	subscribe	Initialize
modbus-write	publish	Enable Unsolicited Events
modbus-write	receiver	Execute Command
other	connect	Upload File
	disconnect	Delete File
	other	other

The sections ahead provide experimental results grouped by the protocols and the respective analysis. It concerns: a comparison between a few Machine Learning settings for each classifier (in order to find the most accurate); metrics evaluation and confusion matrix (in order to evaluate the classifier accuracy for a specific user action). Also, a comment on the classifiers performance and a comparison with the corresponding results.

11.1 Modbus

11.1.1 Random Forest Classifier

The results obtained from the Random Forest algorithm can be seen in the table 11.2, the metrics presented by the means of precision, recall and F1 score. Confusion matrix can also be seen in the figure 11.1. It is clearly evident that all the user actions are correctly classified, and the classification metrics also predict very good results; however, there are some wrongly classified and scored the lowest metrics, these actions are wrongly classified and are denoted as “other” in the field.

Action	Precision	Recall	F1 Score
modbus-read	1	1	1
modbus-write	1	0.37	0.54
other	0.68	1	0.81

TABLE 11.2: Modbus random forest classification metrics

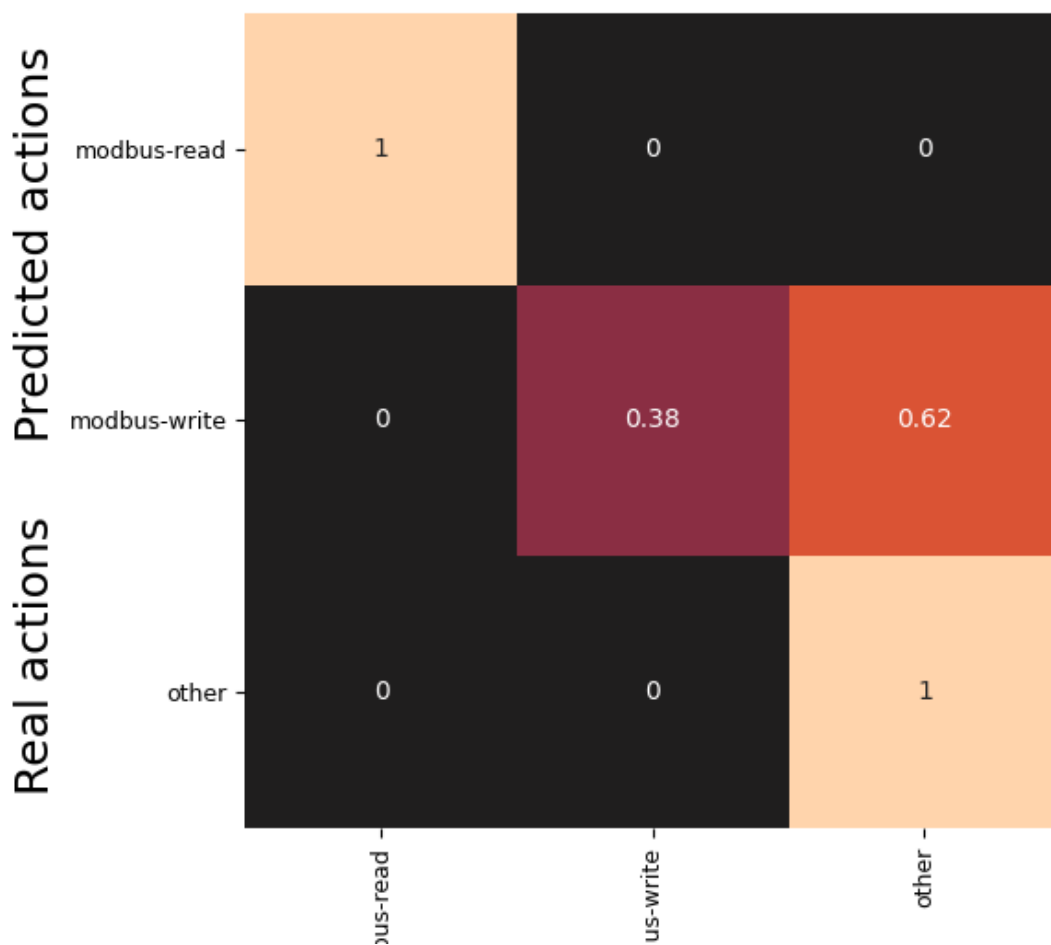


FIGURE 11.1: Modbus Confusion Matrix

11.1.2 Support Vector Machine Classifier

It is worth mentioning that the The result for Support Vector Machine can be classified into three categories; 1) Linear kernel, polynomial kernel and Radial Basis Function (RBF) (also known as the Gaussian kernel) kernel. For generalization and simplicity, I have deployed the linear support vector machine. The table below 11.3 are the obtained results for user actions for the three protocols.

Action	Precision	Recall	F1 Score
modbus-read	0.67	1	0.8
modbus-write	1	0.45	0.6
other	0.6	1	0.75

TABLE 11.3: Modbus SVM classification metrics

In my experiment for the SVM, I have also tuned the C parameter which supports the optimization and tells the SVM how to avoid the misclassification and for each training sample.

The chosen parameters of C are as follows; $C = [0.001, 0.01, 0.1, 1, 10, 100]$. The larger the value of C the smaller the margin in the separating hyperplane, and a smaller value of C will result in encouraging large margin, as a result, the value C acts in SVM as regularization parameter and a trade off of a correct classification.

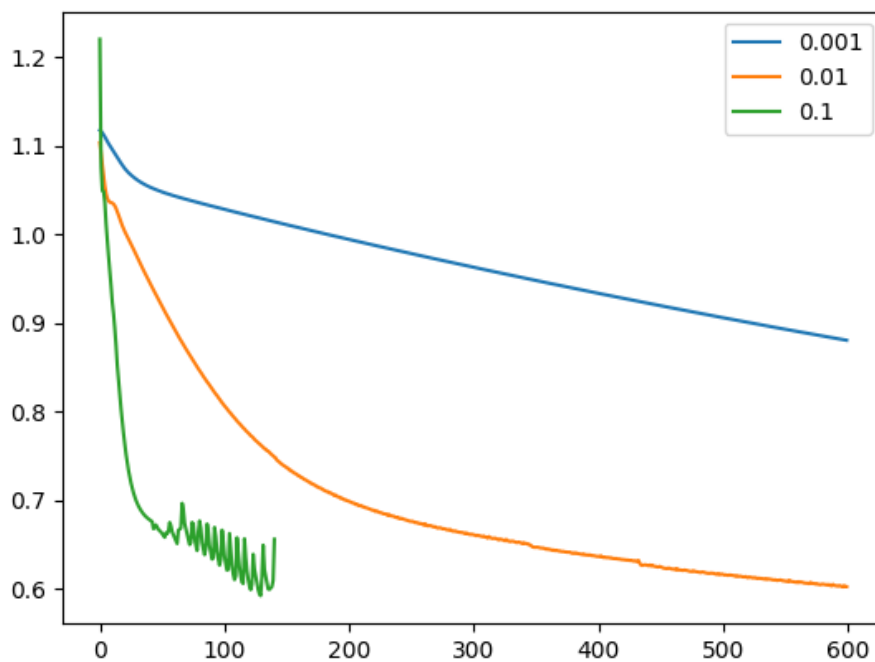


FIGURE 11.2: Learning Curve for Modbus protocol

Notice that in the figure above 11.2 choosing the suitable learning rate for C value can be very crucial. First of all, it's clearly evident that all the three (blue, orange and green) learning rates started almost at the same initial value which is 1.2.2. The small decay value of the blue has the least effect, decreased to the value between 1.1.1 to 1 from 100 to the last 600 iterations. The second change to the learning rate in the orange reduced below 0.7 at almost 200 iterations to close to 0.4 at 300 to the last iteration. Finally, the Green has a dramatic effect, reducing the learning rate to 0.7 within almost 50 numbers of iteration and terminating at around 180 iteration.

In my implementation, the best learning rate is the orange with accuracy of more than 79%. Also it's worth mentioning that all the three learning rates are non-linear with a little fluctuation in the green.

To conclude the discussion on the learning curve, whenever choosing very small learning rate: the optimization will be stable but the convergence can be very slow, while, on the other end, choosing a very large learning rate: the convergence will occur very fast but the optimization will be very unstable. Therefore, the best parameter set found in implementation is ' C ': 0.01.

11.1.3 Deep Neural Network Classifier

The Deep neural network algorithm predicts acceptable results, and a perfect on Recall metric for the modbus-read. The overall results can be visualized on the table 11.4 below.

Action	Precision	Recall	F1 Score
modbus-read	0.67	1	0.92
modbus-write	0.99	0.45	0.6
other	0.8	0.57	0.75

TABLE 11.4: Modbus Deep Neural Network classification metrics

11.2 MQTT

11.2.1 Random Forest Classifier

For the MQTT protocol, I have applied the same methodologies as in the Modbus protocol, however, since we have different actions in this protocol, different results have been obtained.

The table 11.5 shows the result obtained for the random forest classifier.

Action	Precision	Recall	F1 Score
subscribe	1	1	1
publish	0.83	0.83	0.8
connect	0.5	1	0.88
disconnect	1	1	0.9
receiver	1	0.83	0.81
other	1	0.2	0.5

TABLE 11.5: MQTT random forest classification metrics

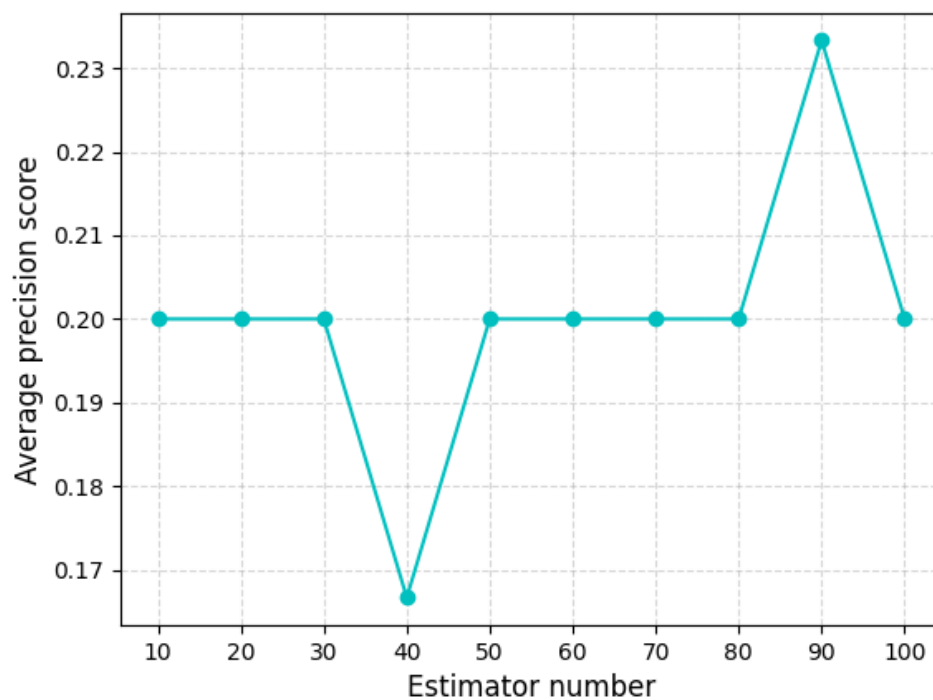


FIGURE 11.3: Precision score for MQTT protocol of random forest classifier in relation to the n estimators parameter.

Notice that, I have also computed the number of estimators for the MQTT protocol. Even though there is no monotonic trend correlating the number of estimators, the best score achieved for the peak is 90 estimators as in the figure 11.3, therefore, this value is chosen for further computations based on this classifier.

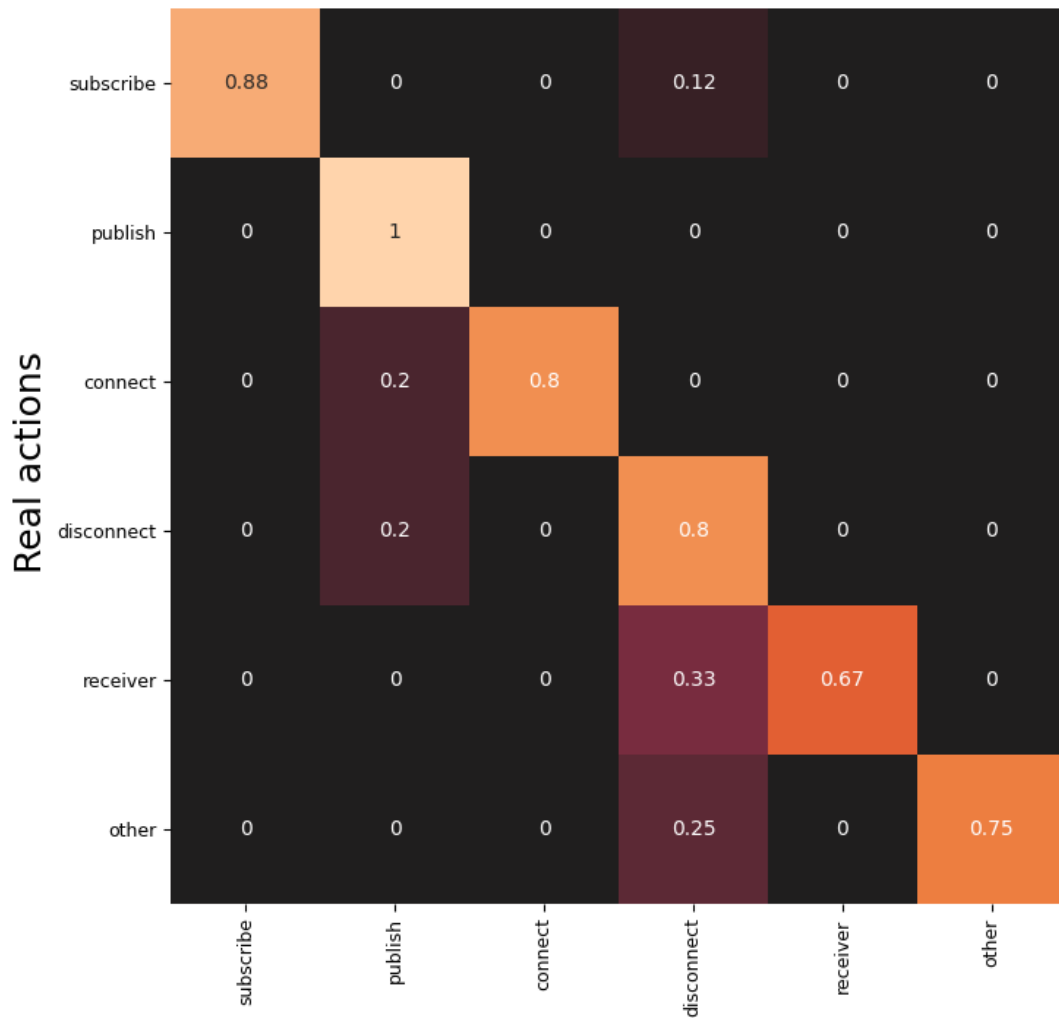


FIGURE 11.4: MQTT Confusion Matrix

The Confusion Matrix in the experiments of MQTT protocol as shown in the figure 11.4 proves that most of the actions taken place have been correctly classified, especially in the “*subscribe*”, “*connect*”, and “*disconnect*” actions.

Be careful in the figure 11.1 not to be confused with the real or actual actions and the predicted actions in the figure of confusion matrix since they are both in the same line to save some space, the matrix of each row in the figure represents the elements in their actual classes “*real actions*” while the matrix of each column representing the elements in as the predicted classes “*predicted actions*”.

11.2.2 Support Vector Machine Classifier

In comparison with Modbus protocol, the classification metrics show that MQTT outperforms the performance of Modbus protocol as demonstrated in the table 11.6 below.

Action	Precision	Recall	F1 Score
subscribe	1	1	1
publish	0.8	0.83	0.8
connect	1	0.8	0.66
disconnect	1	0.8	0.9
receiver	1	0.83	0.9
other	1	0.8	0.5

TABLE 11.6: MQTT Support Vector Machine classification metrics

The best parameter set found in implementation of SVM is ' C ' : 0.10. The figure 11.5 demonstrates the learning curve of the C parameter.

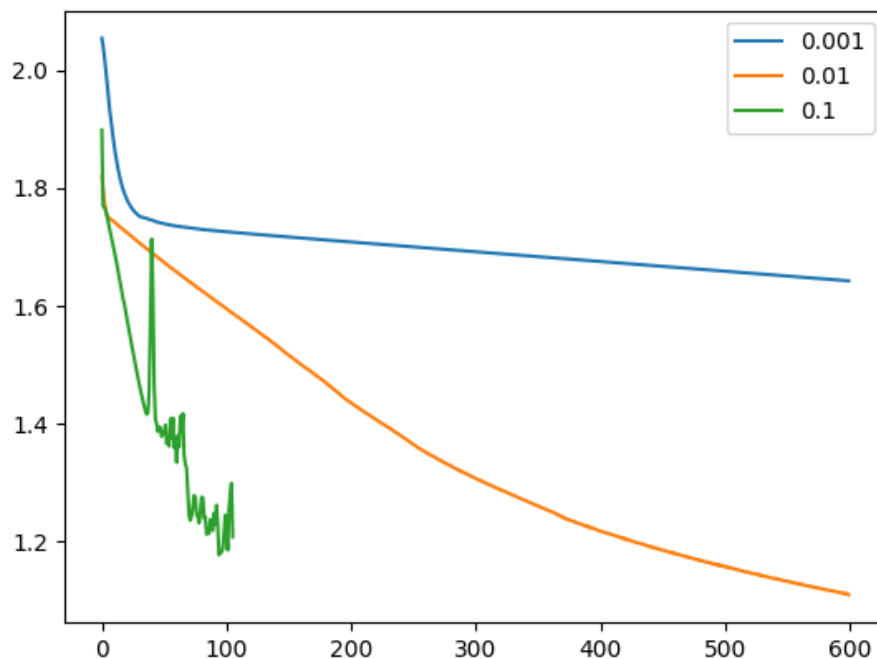


FIGURE 11.5: Learning Curve for MQTT protocol

11.2.3 Deep Neural Network Classifier

The result for Deep neural network classification metrics predicts a similar results trend to the Random Forest classification metrics. The overall results can be seen in the table 11.7 below.

Action	Precision	Recall	F1 Score
subscribe	1	1	1
publish	0.8	0.83	1
connect	0.9	0.8	0.99
disconnect	1	0.8	0.8
receiver	0.86	0.14	0.9
other	1	0.83	0.9

TABLE 11.7: MQTT Deep Neural Network classification metrics

11.3 DNP3

11.3.1 Random Forest Classifier

The table 11.8 displays the classification metrics for the Random Forest classifier of the DNP3 protocol.

Action	Precision	Recall	F1 Score
initialize	0.6	1	1
Enable Unsolicited Events	0.3	0.22	0.8
Execute Command	1	0.8	0
Upload File	0.4	1	0.88
Delete File	1	0.2	0.6
other	0.75	0	0.3

TABLE 11.8: DNP3 Random Forest classification metrics

Overall, the correlation is not strongly verified for some metric of *Recall* and *F1 Score*.

As displayed in the figure 11.6, the best score achieved on its peak was when the number of estimator is equal to 40, 60 to 70 and 90 to 100, therefore estimator number 40 was chosen based on this classifier for further computations.

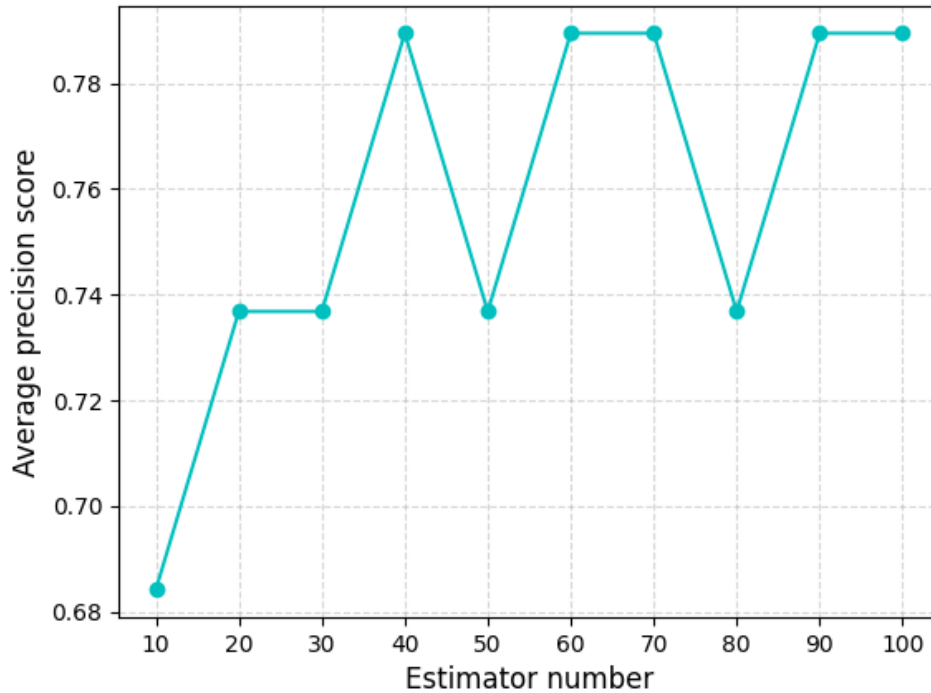


FIGURE 11.6: Precision score for DNP3 protocol of random forest classifier in relation to the n estimators parameter.

11.3.2 Support Vector Machine Classifier

The results for Support Vector Machine classification metrics predicted are somehow different from the Random Forest metrics as the SVM correctly classified all the metrics as shown in the table 11.9.

Action	Precision	Recall	F1 Score
initialize	0.99	0.25	0.75
Enable Unsolicited Events	0.8	0.83	0.8
Execute Command	0.9	0.8	0.7
Upload File	0.98	0.8	0.9
Delete File	0.86	0.14	0.9
other	1	0.5	0.77

TABLE 11.9: DNP3 Support Vector Machine classification metrics

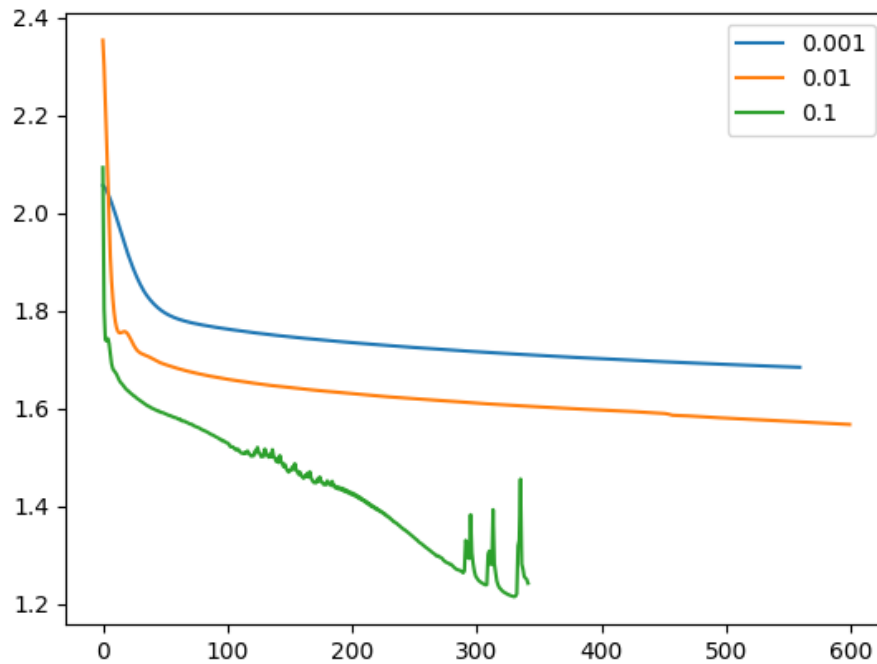


FIGURE 11.7: Learning Curve for DNP3 protocol

The figure 11.8 shows the confusion matrix of all user action and as we can see the figure correctly classified the classes correctly with a very good prediction except for the “*Enable Unsolicited Events*” with 50% and “*others*” with also 50% classified.

The learning curve for Modbus and MQTT protocols converged at around 150 and 110 number of iteration respectively when the learning rate for the value of C is equal to 0.1. for the DNP3 protocol using the same learning rate, we can see that in the figure 11.7 the convergence occurred at more than 330 iteration, however, the best learning predicted for the DNP3 is when the value of C is equal to 10.

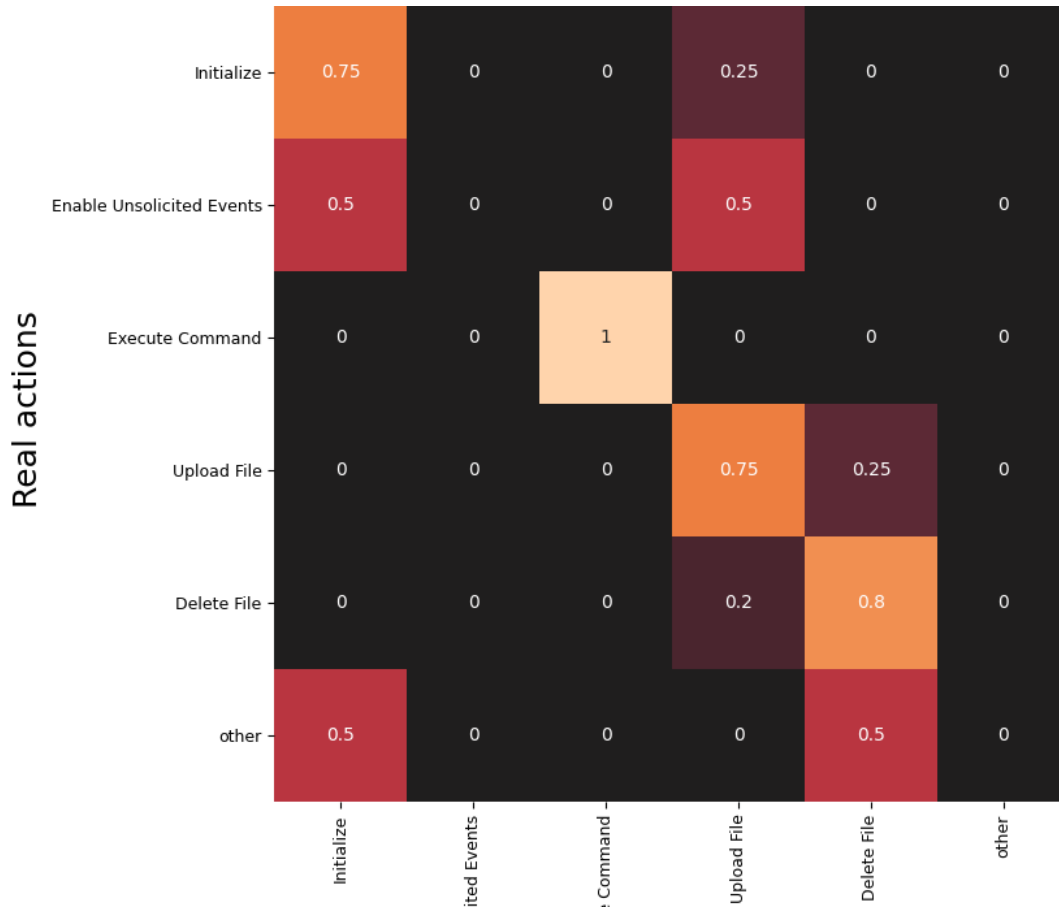


FIGURE 11.8: DNP3 Confusion Matrix

11.3.3 Deep Neural Network Classifier

The Deep Neural Network classifier metrics tend to achieve very good performance for the actions, especially in the “Execute Command” and “Upload File” action as it achieved precision of 100%. The table 11.10 illustrates the results for the performance of the DNN for the DNP3 protocol.

Action	Precision	Recall	F1 Score
initialize	0.88	1	0.33
Enable Unsolicited Events	0.8	0.83	0
Execute Command	1	0.8	1
Upload File	1	0.25	0.75
Delete File	0.86	0.14	0.9
other	0.83	0.33	0.5

TABLE 11.10: DNP3 Deep Neural Network classification metrics

11.4 Performance Analysis and Comparison of ML Algorithms.

11.4.1 Precision Metric

Protocol	Random Forest	SVM	Neural Network
Modbus	0.89	0.75	0.89
MQTT	0.88	0.96	0.92
DNP3	0.67	0.92	0.89

TABLE 11.11: Precision Average Results for Machine Learning algorithms

11.4.2 Recall Metric

Protocol	Random Forest	SVM	Neural Network
Modbus	0.79	0.81	0.73
MQTT	0.81	0.84	0.73
DNP3	0.53	0.55	0.55

TABLE 11.12: Recall Average Results for Machine Learning algorithms

11.4.3 F1 Score

Protocol	Random Forest	SVM	Neural Network
Modbus	0.78	0.71	0.71
MQTT	0.81	0.79	0.93
DNP3	0.82	0.8	0.69

TABLE 11.13: F1 Score Average Results for Machine Learning algorithms

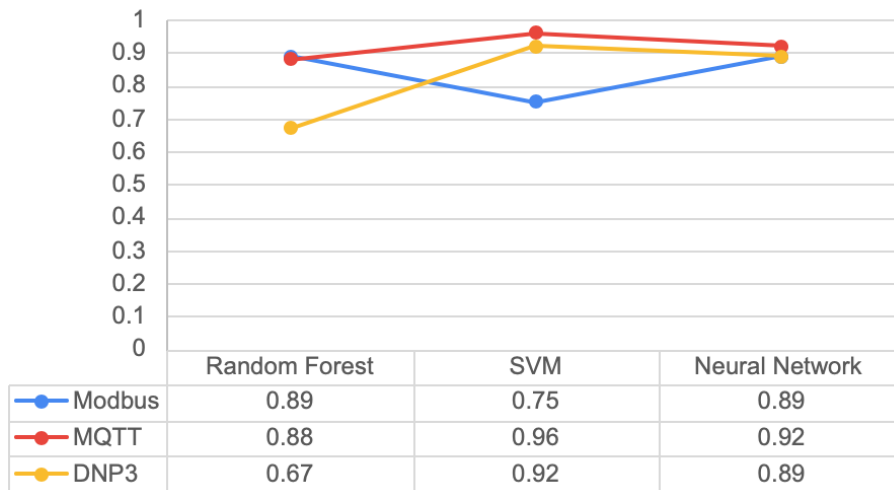


FIGURE 11.9: Precision Performance Analysis

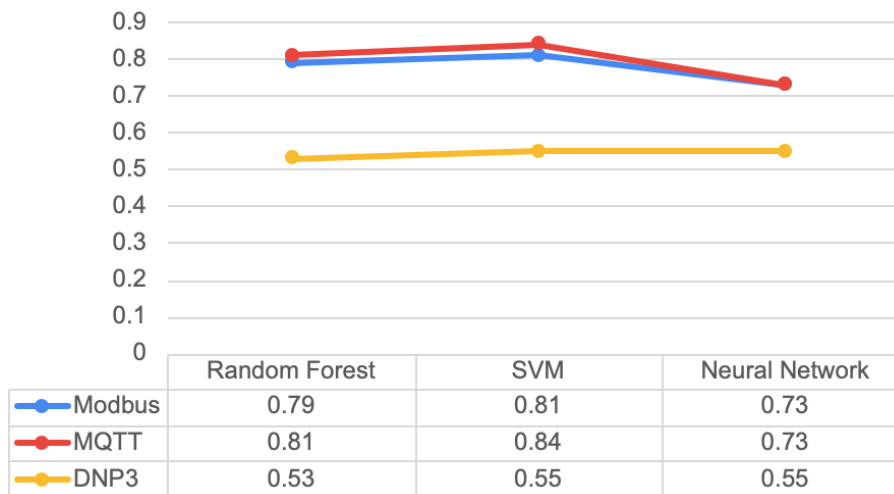


FIGURE 11.10: Recall Performance Analysis

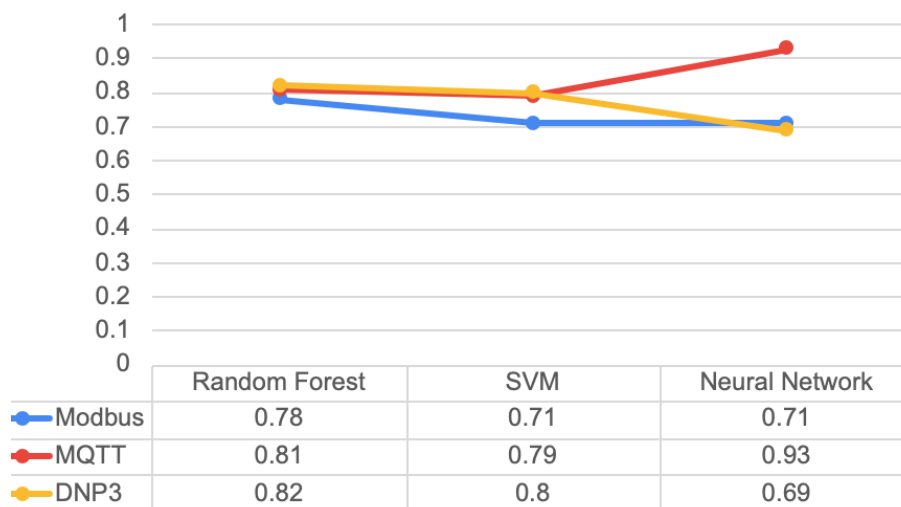


FIGURE 11.11: F1 Score Performance Analysis

11.4.4 summary

The tables presented in the table 11.11, table 11.12 and the table 11.13 shows the average of the result obtained from the classification metrics of the considered machine learning algorithms. Furthermore, the charts given in the figure 11.9, figure 11.10 and the figure 11.11 evaluate the performance of these algorithms in terms of precision, recall and f1 score.

The results obtained for the Random Forest, Support Vector Machine and the Deep Neural Network algorithms are computed as the average of all the entire actions presented in the corresponding protocols.

For the precision metric, it's clearly evident that the Support vector machine algorithm has achieved higher accuracy, specifically in the precision of the MQTT protocol with percentage 0.96%, followed by DNP3 protocol with percentage 0.92% and finally Modbus protocol with 0.75%. The Random Forest and the Deep neural network have achieved almost identical results except that the DNN has better accuracy in the DNP3 and MQTT protocol.

For Recall metric, also Support Vector Machine outperformed the DNN and random forest algorithms for the MQTT protocol and for the Modbus protocol as well. The DNP3 protocol on the other end has achieved the lowest performance with percentage 0.55% accuracy.

Finally, For F1 score metric, The deep neural network outperformed the rest of the algorithms for the actions presented in the MQTT protocol with a percentage of 0.93%. The next best performance is the Random Forest algorithm with percentage 0.81%, while the lowest again is for the deep neural network for the actions presented in the DNP3 protocol. While on the other hand, the lowest accuracy was for the DNP3 protocol of the deep neural network. For simplicity of the visualization, the chart as a graphical representation was provided above.

Chapter 12

Conclusion and Future work

This thesis comprehensively analyzed various aspects of Industrial Control Systems, empirically investigated security issues on ICS protocol and practically implemented encryption on the network traffic and evaluated the performance of machine learning algorithms and analyzed the user actions on the encrypted network traffic.

The main contribution In this thesis is the implementation of the encryption of the Industrial control system protocols and the prediction of user actions in the encrypted network traffic using machine learning algorithms. As we already know that the ICS protocols in their original version are not normally encrypted. Which can lead to serious damage in case an attacker has access to the network traffic. Such behavior can leak information about the communication, as a result, an attacker can easily eavesdrop on the communication and infer the type of the type of the protocol used, the topologies and other sensitive information.

Even though the The classical machine learning techniques have some disadvantages, they are the most trustworthy in comparison to deep learning algorithms, since the classical machine learning algorithms are opaque and less complex. Therefore, in this thesis, machine learning was implemented to support the defence of the security issues and privacy attacks in the encrypted traffic that are accelerated in order to ensure robustness and the accuracy in the model classification.

In the future work, it is possible to consider different types of the ICS protocols since the considered protocols in this thesis are only Modbus, MQTT and DNP3 protocols. And also it is possible to consider various types of machine learning apart from the Random Forest, Support Vector Machine and Deep Neural Network for testing and evaluating the model. These algorithms may include but are not limited to; decision tree, Naive Bayes and k-nearest neighbors algorithm.

Bibliography

- Amoah, Raphael, Seyit Camtepe, and Ernest Foo (2016). "Formal modelling and analysis of DNP3 secure authentication". In: *Journal of Network and Computer Applications* 59, pp. 345–360. ISSN: 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2015.05.015>. URL: <https://www.sciencedirect.com/science/article/pii/S1084804515001228>.
- Andy, Stanford-Clark and Nipper Arlen (2014). "Message Queuing Telemetry Transport Protocol, OASIS Standard". In: ed. by Andrew Banks and Rahul Gupta. London: Palgrave Macmillan UK. DOI: [10.1057/9780230280823_22](https://doi.org/10.1057/9780230280823_22). URL: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.
- Atat, Rachad et al. (2018). "Big Data Meet Cyber-Physical Systems: A Panoramic Survey". In: *IEEE Access* 6, pp. 73603–73636. DOI: [10.1109/ACCESS.2018.2878681](https://doi.org/10.1109/ACCESS.2018.2878681).
- Atkeson, Andrew and Patrick J. Kehoe (2001). *The transition to a new economy after the Second Industrial Revolution. Working Paper 8676, National Bureau of Economic Research*.
- Atterer, Richard, Monika Wnuk, and Albrecht Schmidt (2006). "Knowing the User's Every Move: User Activity Tracking for Website Usability Evaluation and Implicit Interaction". In: *Proceedings of the 15th International Conference on World Wide Web. WWW '06*. Edinburgh, Scotland: Association for Computing Machinery, pp. 203–212. ISBN: 1595933239. DOI: [10.1145/1135777.1135811](https://doi.org/10.1145/1135777.1135811). URL: <https://doi.org/10.1145/1135777.1135811>.
- Bernieri, Giuseppe, Mauro Conti, and Federico Turrin (Nov. 2019). "King-Fisher: an Industrial Security Framework based on Variational Autoencoders". In: pp. 7–12. ISBN: 978-1-4503-7011-0. DOI: [10.1145/3362743.3362961](https://doi.org/10.1145/3362743.3362961).
- Chakure, Afroz (2019). "Random Forest Regression". In: URL: <https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f>.
- Charles, Morris R. and J. E Morris (2012). *The Dawn of Innovation The First American Industrial Revolution*. New York.
- Chen, Thomas M. and Saeed Abu-Nimeh (2011). "Lessons from Stuxnet". In: *Computer* 44.4, pp. 91–93. DOI: [10.1109/MC.2011.115](https://doi.org/10.1109/MC.2011.115).
- Cherepanov, Anton and Robert Lipovsky (2017). "Industroyer: Biggest threat to industrial control systems since Stuxnet". In: URL: <https://www.welivesecurity.com/2017/06/12/industroyer-biggest-threat-industrial-control-systems-since-stuxnet/>.
- Clark, Gregory (2010). "Industrial Revolution". In: *Economic Growth*. Ed. by Steven N. Durlauf and Lawrence E. Blume. London: Palgrave Macmillan UK, pp. 148–160. DOI: [10.1057/9780230280823_22](https://doi.org/10.1057/9780230280823_22). URL: https://doi.org/10.1057/9780230280823_22.

- Clarke, Gordon R. (2004). *Practical modern SCADA protocols DNP3, 60870.5 and related systems*. eng. London.
- Collantes, M. H. and A. L. Padilla (2015). Spain: Spanish National Institute for Cyber-security. URL: https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe_protocol_net_security_ics.pdf.
- Conrad, Eric, Seth Misener, and Joshua Feldman (2017). "Chapter 7 - Domain 7: Security operations". In: *Eleventh Hour CISSP® (Third Edition)*. Ed. by Eric Conrad, Seth Misener, and Joshua Feldman. Third Edition. Syngress, pp. 145–183. ISBN: 978-0-12-811248-9. DOI: <https://doi.org/10.1016/B978-0-12-811248-9.00007-3>. URL: <http://www.sciencedirect.com/science/article/pii/B9780128112489000073>.
- Conti, Mauro, Luigi Mancini, et al. (Mar. 2015). "Can't You Hear Me Knocking: Identification of User Actions on Android Apps via Traffic Analysis". In: DOI: [10.1145/2699026.2699119](https://doi.org/10.1145/2699026.2699119).
- Conti, Mauro, Luigi Vincenzo Mancini, et al. (2016). "Analyzing Android Encrypted Network Traffic to Identify User Actions". In: *IEEE Transactions on Information Forensics and Security* 11.1, pp. 114–125. DOI: [10.1109/TIFS.2015.2478741](https://doi.org/10.1109/TIFS.2015.2478741).
- CRAFTS, NICHOLAS (2011). "Explaining the first Industrial Revolution: two views". In: *European Review of Economic History* 15.1, pp. 153–168. DOI: [10.1017/S1361491610000201](https://doi.org/10.1017/S1361491610000201).
- Darwish, Ihab, Obinna Igbe, and Tarek Saadawi (Jan. 2016). "Vulnerability Assessment and Experimentation of Smart Grid DNP3". In: *Journal of Cyber Security and Mobility* 5, pp. 23–54. DOI: [10.13052/jcsm2245-1439.513](https://doi.org/10.13052/jcsm2245-1439.513).
- Duc, Thang Le et al. (Sept. 2019). "Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey". In: *ACM Comput. Surv.* 52.5. ISSN: 0360-0300. DOI: [10.1145/3341145](https://doi.org/10.1145/3341145). URL: <https://doi.org/10.1145/3341145>.
- Fitzsimmons, Joe (May 1994). "Information Technology and the Third Industrial Revolution". In: *Electron. Library* 12.5, pp. 295–297. ISSN: 0264-0473. DOI: [10.1108/eb045307](https://doi.org/10.1108/eb045307). URL: <https://doi.org/10.1108/eb045307>.
- Franceschett, A. L. et al. (2019). "A Holistic Approach - How to Achieve the State-of-art in Cybersecurity for a Secondary Distribution Automation Energy System Applying the IEC 62443 Standard". In: *2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, pp. 1–5. DOI: [10.1109/ISGT-LA.2019.8895368](https://doi.org/10.1109/ISGT-LA.2019.8895368).
- Geiger, M. et al. (2020). "An Analysis of Black Energy 3, Crashoverride, and Trisis, Three Malware Approaches Targeting Operational Technology Systems". In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1, pp. 1537–1543. DOI: [10.1109/ETFA46521.2020.9212128](https://doi.org/10.1109/ETFA46521.2020.9212128).
- Gollmann, Dieter and Marina Krotofil (Mar. 2016). "Cyber-Physical Systems Security". In: vol. 9100, pp. 195–204. ISBN: 978-3-662-49300-7. DOI: [10.1007/978-3-662-49301-4_14](https://doi.org/10.1007/978-3-662-49301-4_14).
- Gordon, Robert J. (Dec. 2000). "Does the "New Economy" Measure Up to the Great Inventions of the Past?" In: *Journal of Economic Perspectives* 14.4,

- pp. 49–74. DOI: [10.1257/jep.14.4.49](https://doi.org/10.1257/jep.14.4.49). URL: <https://www.aeaweb.org/articles?id=10.1257/jep.14.4.49>.
- Habibi Lashkari, Arash (Aug. 2018). “CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is a network traffic Bi-flow generator and analyser for anomaly detection. <https://github.com/ISCX/CICFlowMeter>”. In: DOI: [10.13140/RG.2.2.13827.20003](https://doi.org/10.13140/RG.2.2.13827.20003).
- Heegard, C. et al. (2001). “High performance wireless Ethernet”. In: *IEEE Communications Magazine* 39.11, pp. 64–73. DOI: [10.1109/35.965361](https://doi.org/10.1109/35.965361).
- Huitsing, Peter et al. (2008). “Attack taxonomies for the Modbus protocols”. In: *International Journal of Critical Infrastructure Protection* 1, pp. 37–44. ISSN: 1874-5482. DOI: <https://doi.org/10.1016/j.ijcip.2008.08.003>. URL: <https://www.sciencedirect.com/science/article/pii/S187454820800005X>.
- Humayed, A. et al. (2017). “Cyber-Physical Systems Security—A Survey”. In: *IEEE Internet of Things Journal* 4.6, pp. 1802–1831. DOI: [10.1109/JIOT.2017.2703172](https://doi.org/10.1109/JIOT.2017.2703172).
- Hussain, M. A. et al. (2016). “DNS Protection against Spoofing and Poisoning Attacks”. In: *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1308–1312. DOI: [10.1109/ICISCE.2016.279](https://doi.org/10.1109/ICISCE.2016.279).
- Kagermann, Henning, Wolfgang Wahlster, and Johannes Helbig (2013). “Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 – Securing the Future of German Manufacturing Industry”. In: URL: http://forschungsunion.de/pdf/industrie_4_0_final_report.pdf.
- Kawano, Kegan and Steve Mustard (2006). “The celebrated maroochy water attack”. In: *Computing Control Engineering Journal* 16.6, pp. 24–25.
- Keti, F. and S. Askar (2015). “Emulation of Software Defined Networks Using Mininet in Different Simulation Environments”. In: *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 205–210. DOI: [10.1109/ISMS.2015.46](https://doi.org/10.1109/ISMS.2015.46).
- Larson, Abdul (2020). “A Review of the Math Used in Training a Neural Network”. In: URL: <https://morioh.com/p/d70aa769173a>.
- Lemaymd (2004). “Distributed Network Protocol 3 (DNP3)”. In: URL: <https://en.wikipedia.org/wiki/DNP3>.
- Liberatore, Marc and Brian Neil Levine (2006). “Inferring the Source of Encrypted HTTP Connections”. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security. CCS '06*. Alexandria, Virginia, USA: Association for Computing Machinery, pp. 255–263. ISBN: 1595935185. DOI: [10.1145/1180405.1180437](https://doi.org/10.1145/1180405.1180437). URL: <https://doi.org/10.1145/1180405.1180437>.
- Liu, Qing and Yingmei Li (2006). “Modbus/TCP based Network Control System for Water Process in the Firepower Plant”. In: *2006 6th World Congress on Intelligent Control and Automation*. Vol. 1, pp. 432–435. DOI: [10.1109/WCICA.2006.1712353](https://doi.org/10.1109/WCICA.2006.1712353).
- Ma, C. Y. T., N. S. V. Rao, and D. K. Y. Yau (2011). “A game theoretic study of attack and defense in cyber-physical systems”. In: *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 708–713. DOI: [10.1109/INFCOMW.2011.5928904](https://doi.org/10.1109/INFCOMW.2011.5928904).

- Marian, M. et al. (2019). "A DNP3-based SCADA Architecture Supporting Electronic Signatures". In: *2019 20th International Carpathian Control Conference (ICCC)*, pp. 1–6. DOI: [10.1109/CarpathianCC.2019.8765963](https://doi.org/10.1109/CarpathianCC.2019.8765963).
- Modbus. Modbus Organization, Inc. Retrieved 2 (2013). "MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3". In: URL: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.
- Nourian, A. and S. Madnick (2018). "A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems Applied to Stuxnet". In: *IEEE Transactions on Dependable and Secure Computing* 15.1, pp. 2–13. DOI: [10.1109/TDSC.2015.2509994](https://doi.org/10.1109/TDSC.2015.2509994).
- Nyasore, O. N. et al. (2020). "Deep Packet Inspection in Industrial Automation Control System to Mitigate Attacks Exploiting Modbus/TCP Vulnerabilities". In: *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 241–245. DOI: [10.1109/BigDataSecurity-HPSC-IDS49724.2020.00051](https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00051).
- Omiccioli, Aldo (2017). "Modbus protocol over RS485 – Part 1 – Introduction". In: URL: <https://web-plc.com/blog/2017/06/01/modbus-protocol/>.
- Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Pessoa, Marcus and Juan Jauregui-Becker (Apr. 2020). "Smart design engineering: a literature review of the impact of the 4th industrial revolution on product design and development". In: *Research in Engineering Design* 31, pp. 1–21. DOI: [10.1007/s00163-020-00330-z](https://doi.org/10.1007/s00163-020-00330-z).
- Peterson, Dale (2017). "Insanely Crowded ICS Anomaly Detection Market". In: URL: <https://www.linkedin.com/pulse/insanely-crowded-ics-anomaly-detection-market-dale-peterson>.
- Petrie, Ian (2007). In: *Technology and Culture* 48.3, pp. 639–641. ISSN: 0040165X, 10973729. URL: <http://www.jstor.org/stable/40061298>.
- Rifkin, Jake G. (2012). "The Third Industrial Revolution : How the Internet , Green Electricity , and 3-D Printing are Ushering in a Sustainable Era of Distributed Capitalism". In:
- Sestito, Guilherme Serpa et al. (2014). "Artificial neural networks and signal clipping for Profibus DP diagnostics". In: *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 242–247. DOI: [10.1109/INDIN.2014.6945515](https://doi.org/10.1109/INDIN.2014.6945515).
- Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press. ISBN: 1107057132.
- Smith, Adam and Alan B. Krueger (2003). *The Wealth of Nations*. Bantam Classics. ISBN: 0553585975.
- Thiel, Frank (2016). "MQTT: Communication in the Internet of Things". In: URL: <https://www.wut.de/e-577ww-05-apus-000.ph>.

- trust, zero (2017). "What is the Purdue Model for ICS Security". In: URL: <https://www.zscaler.com/resources/security-terms-glossary/what-is-purdue-model-ics-security>.
- Upadhyay, Yuvraj, Amol Borole, and D. Dileepan (2016). "MQTT based secured home automation system". In: *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pp. 1–4. DOI: [10.1109/CDAN.2016.7570945](https://doi.org/10.1109/CDAN.2016.7570945).
- Velez, Joshua et al. (2018). "IEEE 1451-1-6: Providing common network services over MQTT". In: *2018 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6. DOI: [10.1109/SAS.2018.8336750](https://doi.org/10.1109/SAS.2018.8336750).
- Watkins, L. et al. (2015). "Using inherent command and control vulnerabilities to halt DDoS attacks". In: *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 3–10. DOI: [10.1109/MALWARE.2015.7413679](https://doi.org/10.1109/MALWARE.2015.7413679).
- Weber, Austin (Apr. 2015). "Assembly Automation Takes Off in Aerospace Industry". In:
- William L. Mostia Jr., P.E. (2019). "Introduction to Modbus". In: 00.00, pp. 1–13. URL: <https://www.controlglobal.com/articles/2019/introduction-to-modbus/?stage=Live>.
- Wyatt, Lee T. (2008). *The Industrial Revolution, 1st Edition*. Greenwood Publishing Group.
- Yampolskiy, M. et al. (2012). "Systematic analysis of cyber-attacks on CPS-evaluating applicability of DFD-based approach". In: *2012 5th International Symposium on Resilient Control Systems*, pp. 55–62. DOI: [10.1109/ISRCS.2012.6309293](https://doi.org/10.1109/ISRCS.2012.6309293).
- Yassein, Muneer Bani et al. (2017). "Internet of Things: Survey and open issues of MQTT protocol". In: *2017 International Conference on Engineering MIS (ICEMIS)*, pp. 1–6. DOI: [10.1109/ICEMIS.2017.8273112](https://doi.org/10.1109/ICEMIS.2017.8273112).
- Yin, Y., K. Stecke, and Dongni Li (2018). "The evolution of production systems from Industry 2.0 through Industry 4.0". In: *International Journal of Production Research* 56, pp. 848–861.
- Yokoi, K. (2014). "Yokoi Style of Sales in Japanese - Akashi City Japan Pencom Publication". In:
- Zanuttigh, Pietro (2020). "Pietro Zanuttigh's slide of the Machine Learning course, class of 2020". In: URL: https://elearning.unipd.it/dfa/pluginfile.php/68775/mod_resource/content/0/22_Clustering.pdf.
- Zhou, C. et al. (2020). "Risk-Based Scheduling of Security Tasks in Industrial Control Systems With Consideration of Safety". In: *IEEE Transactions on Industrial Informatics* 16.5, pp. 3112–3123. DOI: [10.1109/TII.2019.2903224](https://doi.org/10.1109/TII.2019.2903224).