T.C.
ALTINBAS UNIVERSITY
Institute of Graduate Studies

Electrical and Computer Engineering

CLASSIFICATION OF NORMAL AND ABNORMAL ACTIVITY IN SCADA SYSTEM USING MULTI-KERNEL SVM

Amer ALMOZANI

Master of Science

Supervisor
Prof. Dr. Oğuz BAYAT

Istanbul, 2021

CLASSIFICATION OF NORMAL AND ABNORMAL ACTIVITY IN SCADA SYSTEM USING MULTI-KERNEL SVM

by

Amer Sabah Khalaf ALMOZANI

Electrical and Computer Engineering

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled CLASSIFICATION OF NORMAL AND ABNORMAL ACTIVITY IN SCADA SYSTEM USING MULTI-KERNEL SVM prepared by Amer ALMOZANI and submitted on (25/08/2021) has been accepted unanimously/by majority of votes for the degree of Master of Science in Electrical and Computer Engineering.

Prof. Dr. Oğuz BAYAT

Supervisor

Thesis Defense Jury Committee Members:

Prof. Dr. Oğuz BAYAT	School of Engineering and Architecture, Altinbaş University	_____
Asst. Prof. Dr. Muhammad ILYAS	School of Engineering and Architecture, Altinbaş University	_____
Asst. Prof. Dr. İzzet Paruğ DURU	Vocational School, Istanbul Gedik University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies:
____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Amer Sabah Khalaf ALMOZANI

Signature

DEDICATION

To all of my family specially, to my heart, my lovely daughter “Hiba”, hopefully to be proud of me.

ACKNOWLEDGEMENTS

To my respected advisor Prof. Oğuz BAYAT and all Altinbaş University's Staff.

ABSTRACT

CLASSIFICATION OF NORMAL AND ABNORMAL ACTIVITY IN SCADA SYSTEM USING MULTI-KERNEL SVM

ALMOZANI, Amer,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Oğuz BAYAT

Date: August/2021

Pages: (50)

SCADA systems or supervision, control and data acquisition systems for its acronym in English, can be defined in simple terms as networks of controllers and computers that are created in order to remotely control and monitor all types of industrial processes In this paper we propose a feature extraction method that extracts the features of normal activity on SCADA, and abnormal activity in order to feed a classification system, the classification scheme that classifies the activity into normal and abnormal in accordance with the data provided by the feature extraction phase.

Keywords: SCADA, SVM, Kernel, Machine learning, Classification

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
IoT	:	Internet of Things
PAN	:	Personal Area Network
FSK	:	Frequency Shift Keying
TDMA	:	Time Division Multiple Access
OFDM	:	Orthogonal Frequency Division Multiplexing

LIST OF SYMBOLS

μ	:	Mobility of electron
λ	:	Wave length
ω	:	Angle frequency

1. INTRODUCTION

1.1 BACKGROUND

SCADA systems or supervision, control and data acquisition systems for its acronym in English, can be defined in simple terms as networks of controllers and computers that are created in order to remotely control and monitor all types of industrial processes [1]. Basically, they use a physical layer quite similar to traditional information systems, however, they differ from them in their priorities, with availability being the most important in control systems [2]. Due to the fact that in recent years these systems have been the target of various attacks, it is necessary to implement security systems that allow the detection of possible attacks in real time. It is intended that in this way an operator is alerted and consequently can act immediately in the event of an imminent case of improper manipulation of the plant.

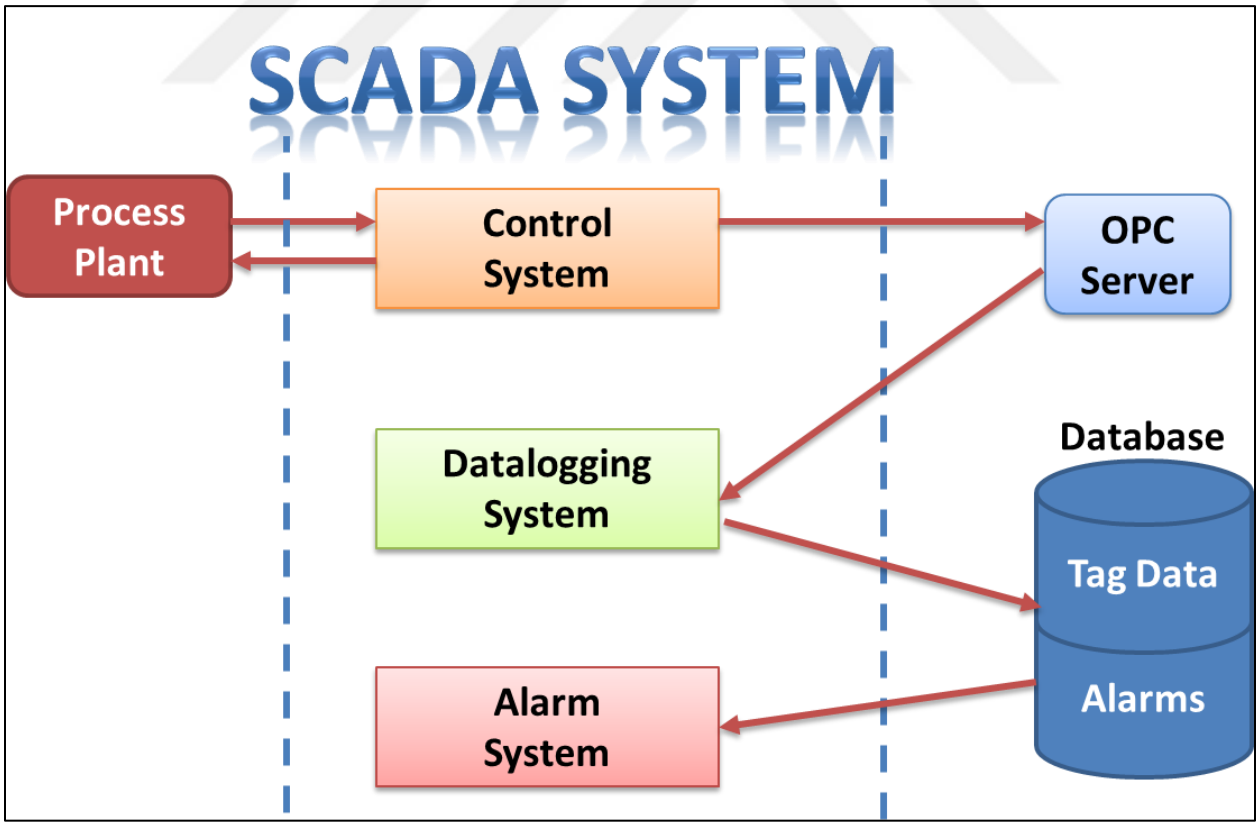

Figure 1.1: SCADA Architecture.

1.2 PROBLEM STATEMENT

A secure computer system must take care to comply with the pillars of cybersecurity, which are: Confidentiality, Integrity and Availability, as shown in Figure 1.2 [4]. Considering that in SCADA systems, the dominant paradigm since its inception has been the availability of information [2], the other aspects have been neglected, giving rise to a scenario where many of the industrial communications lack encryption and as if it were not enough supervision and control equipment run under obsolete software versions, highly vulnerable to all kinds of malware.

1.3 MOTIVATION

Although several techniques and algorithms have been reported in the literature for intrusion detection; most of them can only identify previously known attack patterns using rule-based methodologies; in other solutions that use machine learning algorithms, these are restricted to a certain type of communication protocol. On the other hand, to the best of our knowledge, all the work carried out focuses on the analysis of network traffic, ignoring the internal behavior of the variables within the control devices and therefore the media effects that these could have on the process. Industrial. In SCADA systems, two types of variables are those that allow remote monitoring and manipulation of a plant. The first type is known as supervisory variables and their purpose is to transport information from the plant to the SCADA client. The second type are known as control variables and their purpose is to be able to execute orders from the client application. The proposed solution is aimed at achieving intrusion detection; For this, it must have the ability to recognize the normal behavior patterns of a SCADA system, in such a way that, by identifying an atypical pattern in the behavior of the variables of the control devices, a possible cyberattack can be recognized

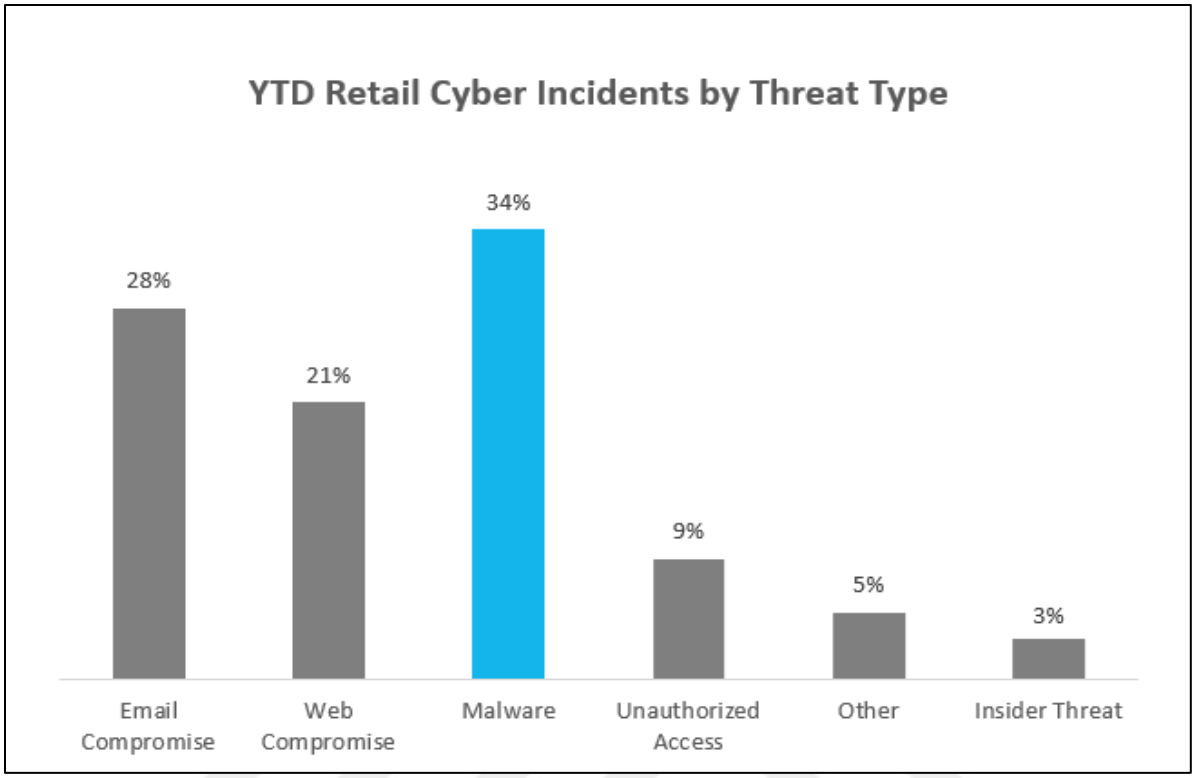

Figure 1.2: Comparison of malware activity to other cyber threats.

1.4 CONTRIBUTION

Malicious content on any network is normally detected through several features, one of the most apparent feature is the extensive exchange in information between users, This exchange of information, called network traffic, is carried out with the transmission of a sequence of Transmission Control Protocol, This traffic can be considered normal or anomalous depending on certain parameters such as the high number of bytes transmitted, or a request for access to an unauthorized machine, etc.

In this thesis we propose a feature extraction method that extracts the features of normal activity on SCADA, and abnormal activity in order to feed a classification system, the classification scheme that classifies the activity into normal and abnormal in accordance with the data provided by the feature extraction phase.

1.5 THESIS ORGANIZATION

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. Section 4 is where we explain our model in details and implementation of that model in details and where we simulate our model and record the results obtained. Section 5 is where we conclude our work and put a future scope into perspective.

2. LITRITAURE REVIEW

This chapter aims to explain some of the previous works done in classifying intrusive malware, looking for viable answers for the current scenario. The objective is to identify unusual behavior that indicates an invasion or use of malicious software during the normal operation of the system.

An article published in 2011with the title of “Adware detection and privacy control in mobile devices” by Ideses et .al, tells that the first computer virus was developed for a demonstration by Leonard Adleman (one of the developers of the algorithm RSA1) at a computer security seminar.

In 2011, Cisco publishes an article in which it says that the use of the internet would quadruple until 2015, driven mainly by the growing number of users of mobile devices (tablets and smartphones). The increase in bandwidth, allowing for higher speeds and ease of access, are relevant contributions to this growth. In this scenario, it is essential to create methodologies that allow a correct classification of network malware. Many network managements tasks, such as workload characterization, capacity planning, route provisioning, network malware modeling and monitoring depends on the identification and classification of that same malware. Today there is a considerable number of applications that use the internet and that number has been increasing steadily. Multimedia applications, social networks, Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) programs, among others, have become essential in the business world, both at a private and corporate level. In this sense, anomalies in network malware can cause serious problems in their operation, being one of the main concerns for entities that depend on the transmission of data for their businesses. The identification and elimination of these anomalies must be done as quickly as possible or there is a risk of high losses as a result of these anomalies. [Jerome et.al] noticed the changes in the characteristics of the network, such as the increase in the transmitted quantity of packets, is one of the specific types of anomalies and can be caused by several factors, such as physical problems or techniques on the network, such as power cuts or inadequate equipment configurations, up to intentional malicious behavior, such as malicious software attacks or unauthorized access attempts.

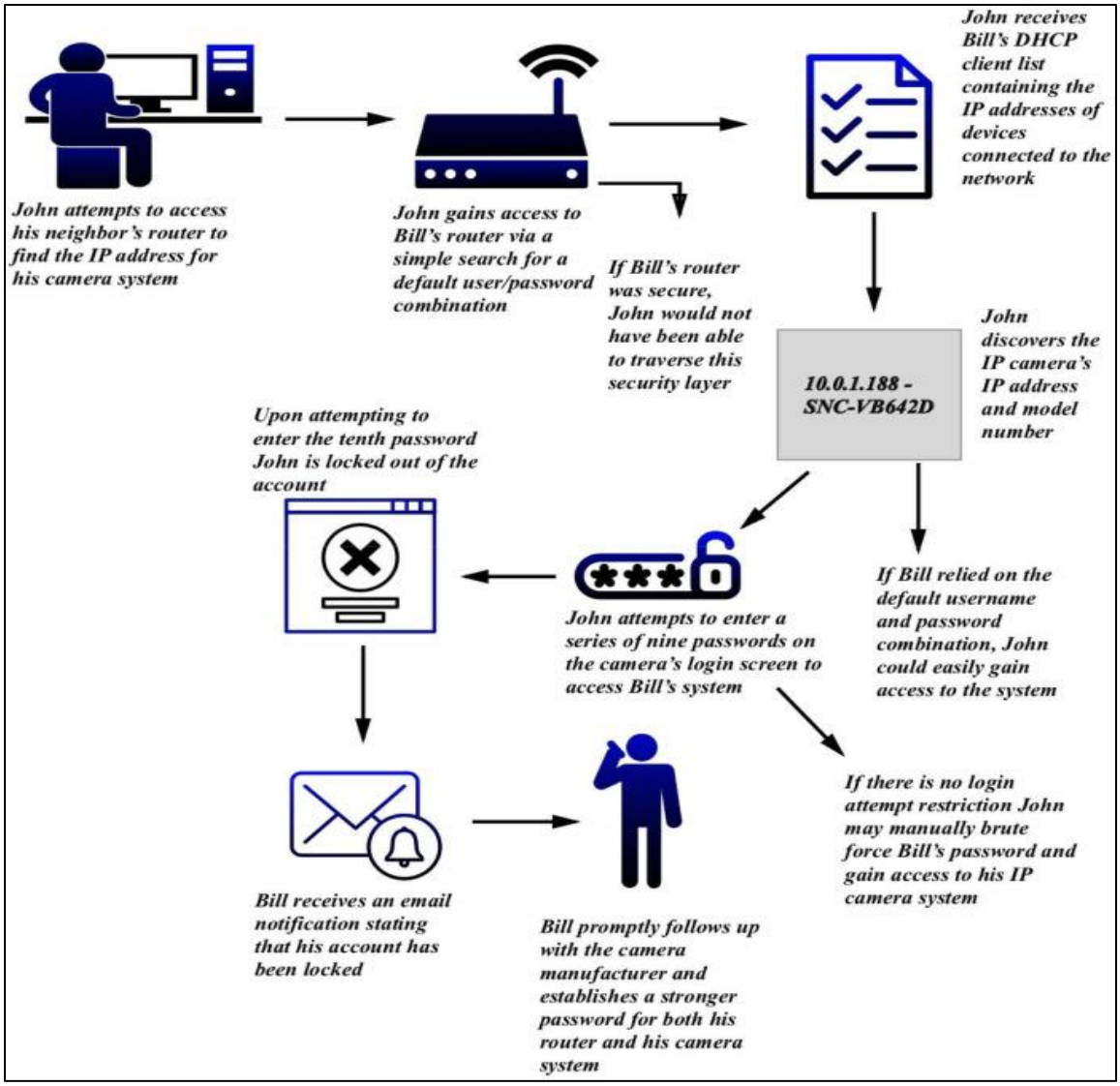

Figure 2.1: CRM DDoS Scenario.

That is why it is essential to find effective network malware classification methods, in order to ensure a quality and at the same time secure service. [Hwan-Hee et.al] explained the main problem in dealing with the analysis of malware on a large-scale network. Techniques for capturing packets and analyzing their contents are generally used to generate statistical data and detect anomalous events. But in a large network that has a considerable number of interconnected devices, this technique becomes unviable, due to the fact that a large processing capacity and space are required to store the data. Traditional classification techniques, such as those based on communication port numbers or packet payload analysis, are also not effective for all types of network malware

[Kurniawan] et.al Due to its nature and the fundamentals of many other techniques, the field of qualification of network malware has maintained a great interest on the part of many authors. For example [Lindorfer et.al] mentions in their work that a common technique for identifying network applications on the Internet is through the monitoring of malware based on usage. to better known communication ports, making an analysis of the packet headers to identify malware that is associated with a particular port and, therefore, of an application in [Mas'ud et.al] This process can lead to inaccurate estimates due to the amount of malware carried out by the most diverse applications, since protocols such as Hyper Text Transfer Protocol Secure (HTTPS), are often used to relay other types of malware , for example, a Virtual Local Area Network (VLAN) over Hyper Text Transfer Protocol (HTTP). On the other hand, some point-to-point applications avoid using known communication ports, in an attempt to circumvent any block imposed by the system, which can cause even more inaccuracies in the estimates. The captured breath, finding similar patterns that can be used for the classification of some anomaly, allows to create a base with data of comparison with the information resulting from this analysis. [Aziz et.al] refers to the concept of anomaly detection using entropy, where Shannon's entropy is cited. This type of detection and classification is done through the entropy calculation for the following categories: source port, destination port, IP address, source and IP address, destination, from the correlation between them. Entropy values can be compared to classify abnormal behaviors in the analyzed malware [Xiaoyan et.al] During the research carried out for this work, it was detected that there is a relationship between the malware class and its statistical properties that provide information on the distribution of the byte flow of packages and the output of a specific number of specific applications.

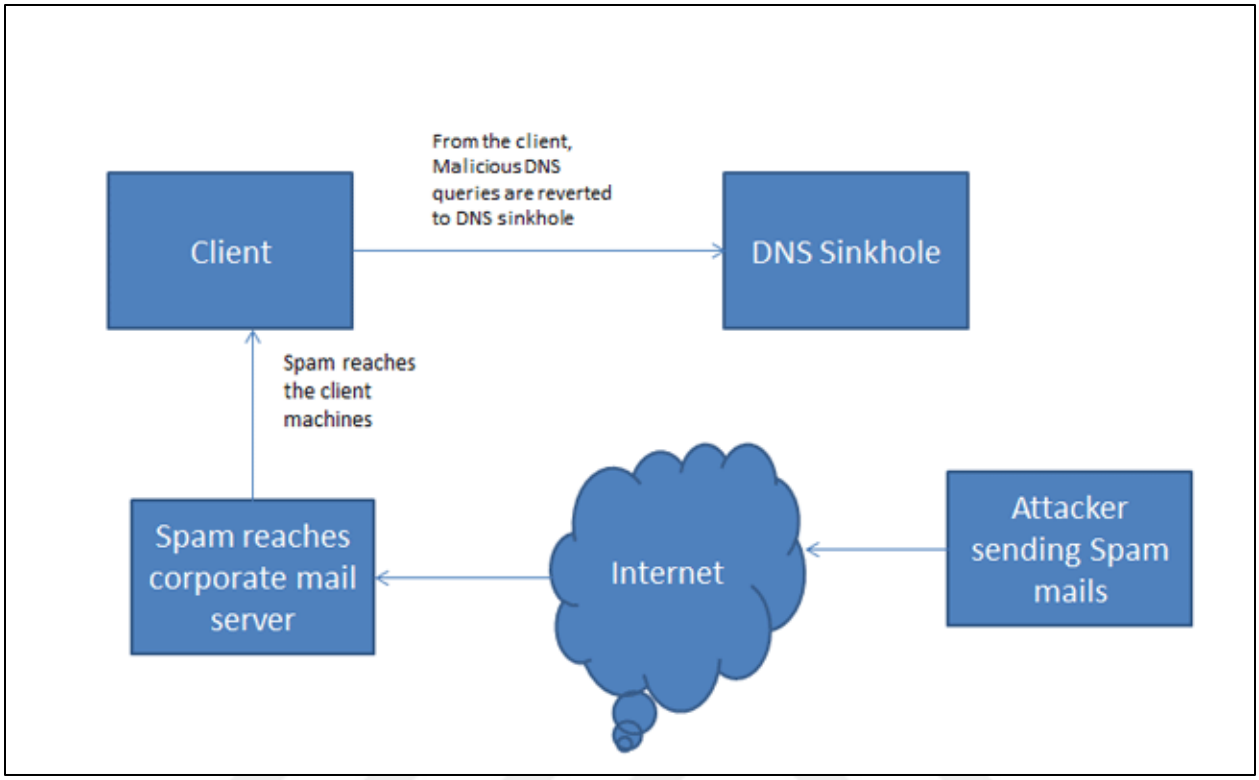

Figure 2.2: Malware that depends on malicious DNS response.

[John et.al] also observed that the relevance of the packages used in the studies carried out by [Westyarian et.al] tend to a certain profile size, the observation of this profile allows a more precise selection of the three relevance to these studies. [Aziz et.al] proposed a classification scheme should allow for the description of representative malware patterns. shows our version of the steps to follow, which serve as guides for carrying out this process. In the area of security, using this information is crucial in identifying anomalies in the network system. Our goal is to understand how the characterization and detection of anomalies in network malware is done.

3. MATERIALS AND METHODS

This chapter describes the necessary and essential concepts in the development of this work, in order to make a classification of the existing malware for a better understanding of the different malware behaviors.

3.1 MALWARE

The creation of malicious software is the same as any other computer application, using its own methodology in its design and implementation. This development methodology is called life cycle and contains a series of steps with the creation of the malware itself the beginning of that cycle. Basically, it can be said that this cycle includes the following steps.

Creation: Most of the time, the creation of malicious software begins with a proposal by individuals with malicious intent with some prop Illicit activity. Generally, the development of this code aims to exploit and take advantage of some vulnerability existing in the invaded system, with the purpose of obtaining financial profits or simply causing damage to the invaded institution.

Development: It is wrong to believe that the creation of malicious software requires knowledge of programming language. There are many tools to create programs capable of carrying out attacks on the system without necessarily being an expert in programming.

Replication: Once the development is finished, some method is used for the transmission.

Usually, the propagation mechanisms chosen reflect the forms of mass communication. Among the most common are electronic mail and P2P networks. Whoever develops the program is not necessarily the same who propagates it.

Data transition: This phase is one of the most important and always obeys the essence of malicious software. This transmission can be activated by a conditional method, when the user performs a specific action or when the equipment clock sets a special date, etc., or directly when the malware starts to implement its functions when the infection is completed. From the moment the code is circulated for its propagation, it is up to the antivirus programmers to detect it before they cause any damage to the user.

So, we have the following cycle:

Characterization: And from that moment on, antivirus companies detect the spread and the characteristics of the code to create their signature (their impression) to digital that serves as a unique identification.

Heuristic detection: This consists of examining and reporting on files that behave like viruses but that are not contaminated with any known code. Heuristic detection looks for certain characteristics in the behavior of an application and tries to determine whether or not it is a malicious code.

Signature detection: After identifying the threat, antivirus programmers analyze the code to find an appropriate detection method and update the signature database. This process must be fast and essential so that the malware is not able to cause damage to users.

Elimination: In this last step, all updates with the signatures corresponding to the recognized malicious codes should already be available to the end user, so that it can be detected and eliminate any type of action caused by the malware.

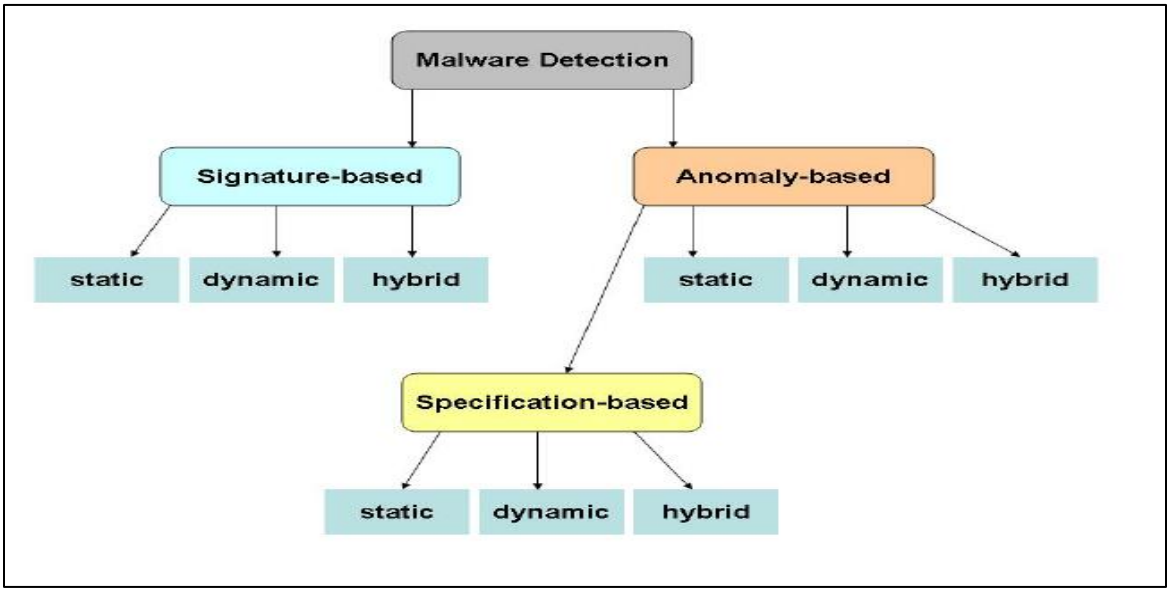

Figure 3.1: Malware detection.

3.2 MALWARE DETECTION

Any software in a computer, is a file containing instructions that will be stored in memory and used by the processor. These instructions are the calls that the software makes to the system when they are executed. The computer and other peripherals have a binary language composed only of 0 and 1. As the computer only understands the binary, the instructions of the software are also represented in binary. Thus, analyzing certain software turns out to be a complicated task. There are some methods that can be used to detect malware that may or may not be used together. Currently, it is common for antivirus manufacturers to use multilayer technology, where files are processed and, when file analysis is unnecessary, it is excluded from the next layer, allowing a faster scan.

Quick Signature-based detection: This method attempts to match files with known malware signatures which is a byte sequence characteristic of specific malicious software. A detailed analysis is needed to identify the exact infection. Detection based on polymorphs It is used to determine new variants of recognized malicious software, even if the behavior of the new variant is different. This method is particularly effective in detecting software that uses macros and scripts. Analysis based on heuristics analysis looks at how software behaves, so it can identify whether it is malicious or not. This allows the detection of unknown malicious software.

Behavior-based analysis: This technology analyzes how the software behaves to determine whether the file's behavior is hostile and prevent its execution.

In the next chapter, we will address the methods used in classifying network traffic to identify anomalies based on this information.

3.3 TYPES OF MALWARE

Computer Virus Appeared for the first time in the 1980s. They presented a simple code, only occasionally causing inconvenience or just showing information to the user. Today it is a type of malware that is developed by programmers that, like a biological virus, infects a system, replicates itself and tries to spread to other computers that are connected to the same network. Currently, this type of code is considered to be of high risk, and it can cause significant damage to the information systems of organizations.

The infection mechanism differs from each virus, the most common being the inclusion of its code in executable programs so that when the infected program is executed, it also runs the program. viral program, thus contaminating other programs.

Viruses can be divided into two broad categories:

Compiled viruses: your source program is compiled and targeted to a specific type of operating system.

Interpreted virus: its source code can only be executed by a specific application or service, such as macros or scripts. The types are as below:

Spyware: Spyware are a code malicious software that gathers information from the user without his knowledge. They differ from trojans because they do not have the objective of controlling infected equipment. They are mainly aimed at stealing users' confidential data such as bank account data, access passwords and documents.

Trojans: Trojans or Trojan horses, as in Greek mythology, is a malicious code that hides an unexpected action, entering a system and opening a communication door to a possible invasion. Currently, this type of code is disguised as a normal application, for example a game, which makes its identification difficult. This type of code is not replicated. The most common functions presented by this code are to collect information, hide its presence and disable security software such as AV, firewall, etc.

Botnets: Botnets are networks made up of thousands of zombie computers, using coordinated attacks, such as Distribution Deny of Service (DDoS), spamming and phishing.

Worms: Worms are similar to viruses, they self-replicate from infected equipment to another automatically, in addition worms have the ability to spread quickly by manipulating the vulnerabilities of a system or application installed on the equipment. The great threat of this type of malicious code lies in the fact that it affects the performance of a system, overloading services and generating traffic, which in some cases causes denial of service attacks (Denial of Service (DOS)).

Backdoor: It is a security flaw that can exist in an application or system that allows unauthorized people to invade, allowing control of the equipment. Generally, they have special abilities.

Zombies: This term comes from the use of machines infected with bots, possessing propagation capabilities like a worm. They are controlled by attackers who make up a larger system.

Remote Administration Tools (RAT): allows the attacker to have access to a system at any time, having access to screen images, keystrokes and files and it is possible to remotely activate devices such as printers, webcams, microphones and speakers.

Rootkit: Rootkits are software, most of the time malicious, that was developed to hide some normal processes or methods of detection and allow privileged access to a system.

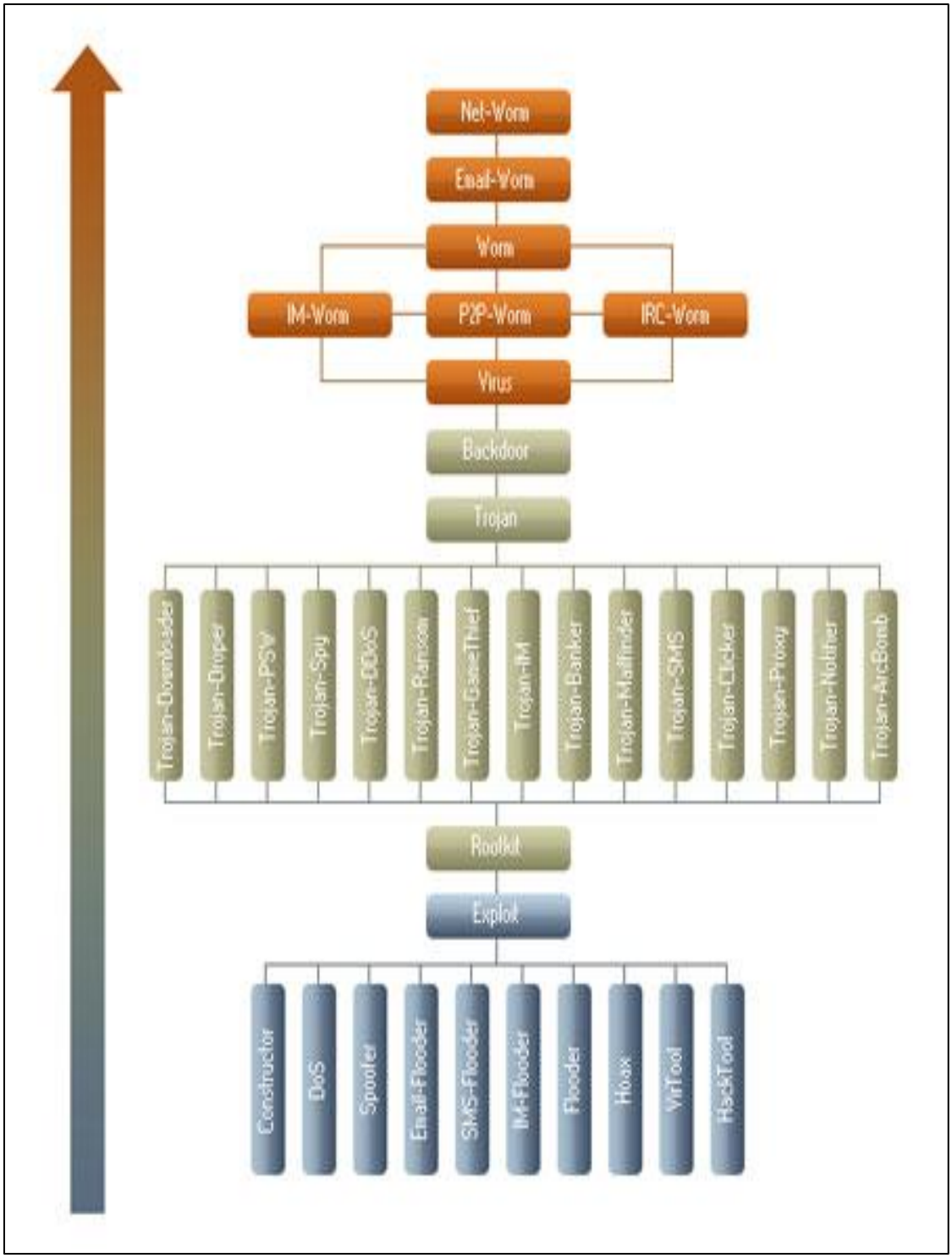

Figure 3.2: Malware risk analysis.

3.4 MALWARE ANALYSIS

The purpose of malware analysis is to obtain essential information to protect a computer network from possible attacks on systems. Its objective is to determine exactly what is happening, alerting those responsible for the system about possible anomalies in the normal functioning of the network Today, with the expansion of communication systems and the development of mobile devices with increasingly high computational characteristics, together with the ease of being connected to the internet at any time and place, it intensifies it of socioeconomic relations worldwide,

a consequence of this technological advance, which has enabled greater economic and cultural integration between people and organizations from different points of the planet, caused people to emerge with the intention of causing damage or taking advantage of the information that users and organizations have stored in their equipment, using malicious code.

```
1  http://www.trustedsite.com/search.html?type=<''<<sCrIpT>alert
2  (document.cookie)</sCrIpT><''<<sCrIpT>alert(document.cookie)
3  </sCrIpT>
4  http://www.trustedsite.com/search.html?type=%3C%22%3C%3C
5  %73%43%72%49%70%54%3E%61%6C%65%72%74%28%64
6  %6F%63%75%6D%65%6E%74%2E%63%6F%6F%6B%69%
7  65%29%3C%2F%73%43%72%49%70%54%3E%3C%22%3C
8  %3C%73%43%72%49%70%54%3E%61%6C%65%72%74%28
9  %64%6F %63%75%6D%65%6E%74%2E%63%6F%6F%6B%
10 69%65%29%3C%2F%73%43%72 %49%70%54%3E
```

Figure 3.3: Example of Malicious code.

Malicious code is understood as a set of applications and other types of programs that aim to compromise the security of a system. This code or program is called malware, which derives from the English words malicious and software and serves to designate any software intended to infiltrate a computer system illicitly, with the unique to cause damage or steal information [5] In order to facilitate the understanding of the behavior of this type of code, it is necessary to classify it to develop a methodology with security policies that help in preventing unwanted action of this code. The most common method used to identify malware is to survey its actions on a compromised system. Their analysis can be divided into two types [6].

Static analysis: which obtains information about the malicious code without having to execute it

Dynamic analysis: where the behavior of the malicious code is monitored during its execution.

In general, malicious codes can be classified according to some specific characteristic of their behavior [7] However, framing a particular piece of malware in a single class is a difficult process to be performed, due to the evolution of the code and the ease of adding new features that make this process difficult. In addition, the new generation of malware has the ability to disable antivirus programs, hiding in the system which makes it even more difficult to identify them.

The taxonomy presented in this document is used to refer to certain types of malware, and identifiers assigned by antivirus mechanisms based on the behavior of the specimens throughout their life cycle. Malicious software can be divided primarily through its dependence or not on the host as described They use communication tools to spread.

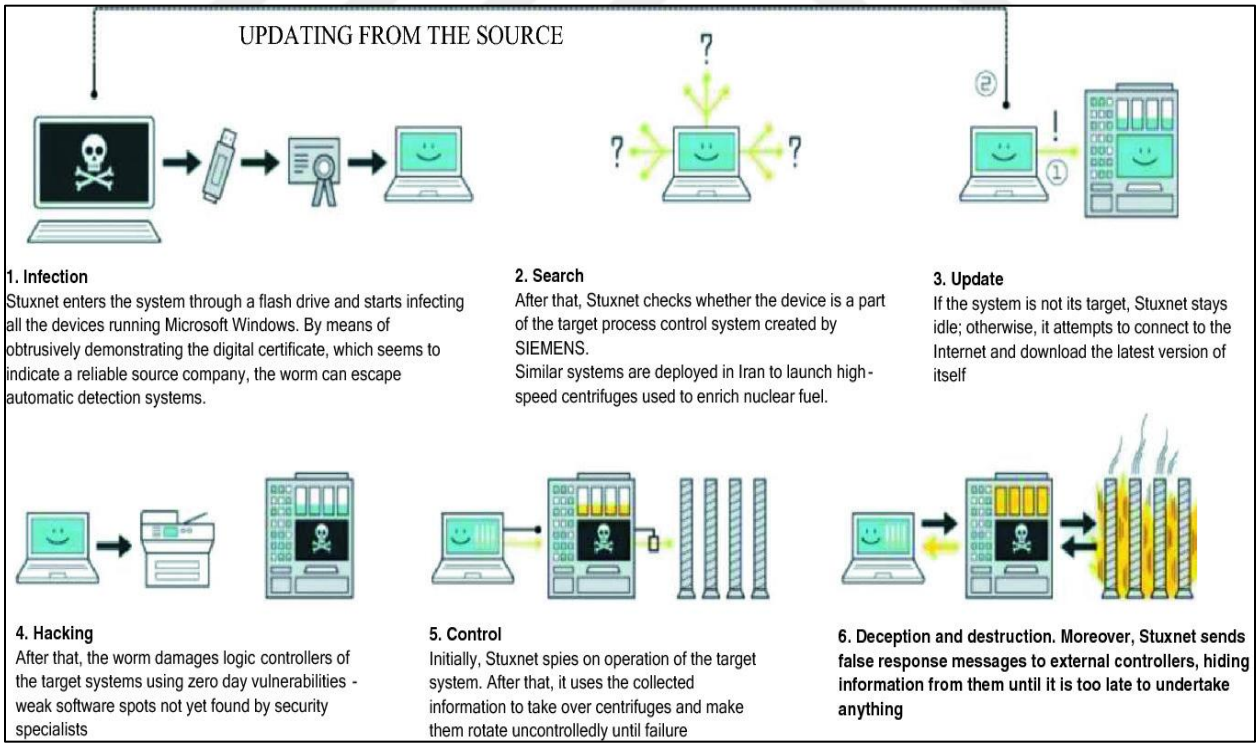

Figure 3.4: Worms workflow and victims without IDS.

Worms, for example, are sent by e-mail or via instant messages, trojans come from websites and viruses are obtained by downloading infected files. It is quite common to find malware that has hybrid capabilities, with characteristics according to your needs. Some strains of malware try to exploit the vulnerabilities existing in the systems, allowing an unauthorized invasion and or more privileges to the system. The intent of any malware is to remain incognito, hiding actively or simply not being noticed in the infected system. More than 90% of the malicious programs that can be found have no description. Most of these programs use names similar to applications and services in operating systems, in order to deceive users. They are usually installed in some system folder and copy themselves to other folders. This ensures that the malware is always active, regardless of who is using the equipment. Malware, as already mentioned, is a software code that aims to exploit some vulnerability, allowing its creator to have access to information contained in the hacked equipment. This type of software can cause effects ranging from an increase in the processes performed on the equipment, causing a considerable decrease in the performance of the attacked equipment, up to a significant increase in the transmission of information.

This is considered an anomaly, which in computer networks typically refer to circumstances where the network's operations differ from the standard considered normal Anomaly detection is defined as the data analysis process with a behavior considered different from the normal behavior expected in a given set to which it belongs. Following this reasoning, the detection of anomalies is defined as the task of determining discrepancies between the behavior found and the behavior expected in the network Intrusion is defined as the set of actions triggered by the attacker, compromising the basic structure of information security, such as the integrity, confidentiality and availability of certain information. Of a computer system. The attacker is defined as any user or entity that invades an unauthorized system, with the intention of provoking illicit acts. Detecting an attacker or intruder is a complex task, since it may have in its power the identification of a legitimate user, or behave as such, and can be classified as internal or external Intrusion detection is the process of detecting unauthorized use or attacks on a computer system or network.

A detection system or Intrusion Detection System (IDS) has the purpose of inspecting the content of network traffic and locating possible attacks, such as Antivirus (AV) software.

It inspects the transmitted content looking for signatures of computer viruses, patterns that correspond to known malicious software or patterns of behavior considered unusual.

The concept of valuation is related to the action and the effect of valuing, which allows to mark, estimate, appreciate or calculate the value of something. An evaluation is a way of quantifying something.

However, finding a way to make an evaluation of malicious software with some consistency is a process with a high level of difficulty. One possibility is to base, for example, on the behavior of the malware, identifying malicious software according to typical actions There are many methods that are used to identify malicious software. These can be the detection of computer viruses, such as worms and trojans, using their signature, with Antivirus (AV) software. Another technique involves behavior-based detection, identifying malicious software in typical actions, such as unsolicited connection and concealing to the Internet, which is usually the case of trojans and keyloggers.

3.5 INTRUSION DETECTION

The creation of malicious software is the same as any other computer application, using its own methodology in its design and implementation. This development methodology is called life cycle and contains a series of steps with the creation of the malware itself the beginning of that cycle. Basically, it can be said that this cycle includes the following steps:

Creation: Most of the time, the creation of malicious software begins with a proposal by individuals with malicious intent with some prop Illicit activity. Generally, the development of this code aims to exploit and take advantage of some vulnerability existing in the invaded system, with the purpose of obtaining financial profits or simply causing damage to the invaded institution. Development: It is wrong to believe that the creation of malicious software requires knowledge of programming language. There are many tools to create programs capable of carrying out attacks on the system without necessarily being an expert in programming. Replication: Once the development is finished, some method is used for the transmission.

Malware forms harmful applications to the infected devices, either for its functioning, capacity or performance. It is important to note that the presence of malware does not necessarily mean the destruction of the equipment. In fact, recent studies indicate that up to 8 out of 10 computers have malware installed and continue with their “almost” normal operations, albeit with some slowness. The virus is generally an automatic performance program, designed to interrupt the operation of the equipment and in some cases, also the destruction of data. It is possible to identify the presence of a virus when there is a change in the normal functioning of the equipment, such as slowness, disappearance of files, or autonomous interruption of its operation. Viruses and other malware can come hidden or encapsulated within content downloaded from the internet. Worm is a type of malicious code that can reproduce itself by occupying space available on the equipment. This will decrease the performance of your computer, without any major consequences, allowing you to identify them or at least suspect their existence. Trojans, in turn, are artifacts that are hidden inside other files (hence their name, in allusion to the Trojan horse). Its main function is to facilitate the theft of the content contained in the equipment such as passwords, files containing personal data, etc.

They have different levels of risk and are identifiable using anti-malware or anti-virus programs. Thus, the main characteristics of a successful IDS are considered the capabilities of:

a. transmission or reception over the network;

b. transmission or reception by e-mail;

c. encapsulation in files downloaded from sites on the network;

d. transmission via social networks;

e. hides in instant messages;

f. exploit system vulnerabilities;

g. closing;

h. deletes files;

i. spends system resources;

j. steal information.

Usually, the propagation mechanisms chosen reflect the forms of mass communication. Whoever develops the program is not necessarily the same who propagates it. Data transition: This phase is one of the most important and always obeys the essence of malicious software. This transmission can be activated by a conditional method, when the user performs a specific action or when the equipment clock sets a special date, etc., or directly when the malware starts to implement its functions when the infection is completed. From the moment the code is circulated for its propagation, it is up to the antivirus programmers to detect it before they cause any damage to the user.

So, we have the following cycle:

Characterization: And from that moment on, antivirus companies detect the spread and the characteristics of the code to create their signature (their impression). to digital) that serves as a unique identification.

Heuristic detection: This consists of examining and reporting on files that behave like viruses but that are not contaminated with any known code. Heuristic detection looks for certain characteristics in the behavior of an application and tries to determine whether or not it is a malicious code.

Signature detection: After identifying the threat, antivirus programmers analyze the code to find an appropriate detection method and update the signature database. This process must be fast and essential so that the malware is not able to cause damage to users.

Elimination: In this last step, all updates with the signatures corresponding to the recognized malicious codes should already be available to the end user, so that it can be detected and eliminate any type of action caused by the malware.

3.6 INTRUSION IN SCADA

A secure computer system must take care of complying with the pillars of cybersecurity, which are: Confidentiality, Integrity and Availability, as shown in Figure 3.5 [4]. Considering that in SCADA systems, the dominant paradigm since its inception has been the availability of information [2], the other aspects have been neglected, giving rise to a scenario where many of the industrial communications lack encryption and as if it were not enough supervision and control equipment run under obsolete software versions, highly vulnerable to all types of malware In the past, SCADA systems were conceived as totally secure environments since in most cases they were totally isolated from access networks public, making an external cyber-attack practically impossible. However, in 2010 a computer worm called Stuxnet attacked a Siemens proprietary SCADA system called WinCC. This worm is recognized for having the ability to reprogram the PLCs and hide the changes made, being also capable of spreading using mass storage devices [5]. The virus not only revealed the vulnerabilities of SCADA systems, but also set off alerts worldwide in the event of a possible cyberattack that would jeopardize critical infrastructures (electricity, chemical industry, etc.) and therefore the security of a company. entire nation [6]. determined the advances in industrial cybersecurity as a priority issue for the protection of the country's infrastructure against a possible cyber war [7]; while DELL's annual cyberattacks, report showed that in 2014 alone, the number of attacks on SCADA systems doubled compared to 2012 [8]. In addition to the threats that can arise within SCADA systems, many of these systems are no longer totally isolated from public networks such as the internet; Therefore, concepts that previously only had validity in traditional telecommunications such as perimeter security, firewalls, intrusion detectors, among others, are now taken into account and are the reason for the development of multiple investigations worldwide, especially in Europe and North America. which will be discussed later. Given their flaw in security issues, SCADA systems become vulnerable, among other types of attacks, to attacks of the "Identity Spoofing" type. In this type of attack, a hacker could tamper with network communications, injecting corrupt commands or packets into the system that the final control elements assume from a trusted operator [9]. While it is true that all types of cyber-attacks produce unwanted effects on the operation of an autonomous system, attacks that involve command injection are even more dangerous, since they directly modify the control parameters of one or more processes, causing them to collapse. Taking into account that in a

SCADA system there are supervision and control variables, an attack that corrupts the information of the supervised variables would result in false information for the operator, which would imply, for example, that the operator would not detect possible anomalies; while an attack that corrupts the information of the controlled variables would result in the malfunction of the actuators (motors, pistons, resistors) almost immediately, and could cause a disaster. At this point, an intrusion detection system, IDS for its acronym in English, which can only detect known attacks or that simply verifies changes in the behavior of the traffic, would not be able to identify some types of intrusion, because to achieve this it requires in advance know what is the behavior of the process variables in each of their different states. In other words, an ideal IDS system should in principle be trained and learn from the system variables what its ideal behavior is, which implies the use of machine learning algorithms, in such a way that it can recognize anomalous behavior patterns by means of the in-depth analysis of the variables regardless of their origin. On the other hand, the reading of the variables directly in the controlling device, allows to determine even the insertion of a possible anomaly, which, although it does not have immediate repercussions, could have catastrophic effects in the long term. Through this work, the approach, design and implementation of a system for the detection of intrusion in controllers associated with a SCADA network is described that allows, through the use of machine learning algorithms, to detect when a corrupt package tries to modify the normal operation of a process. This in order to alert the operators about a possible attack and that they can take the pertinent measures as the case may be.

3.7 MACHINE LEARNING

Machine Learning originated in the field of Artificial Intelligence. It forms a set of mathematical and statistical models whose tasks involve recognition, diagnosis, robot control, prediction, etc. Machine Learning is used in data science as a tool, because it produces autonomous results without the need to be explicitly programmed, having gone through a training stage [14]. A traditional example of Machine Learning is a spam mails prediction system, where the input is a set of mails that have been labeled as spam by humans. The learning process consists of recognizing a set of words and characteristics that appear in a spam mail. Then, when a new mail is received, its characteristics will be reviewed and if it fits into the set of suspicious content, it will be labeled as spam mail. This system, with an adequate training database, is able to correctly predict the label of all new incoming mail. The field of Machine Learning has been divided into a few sub fields to deal with the different types of learning tasks. For research objectives, emphasis will be placed on the pair: Supervised and Unsupervised.

3.7.1. Unsupervised Learning

In this case, there is no distinction between training and test data, only input data is needed for the algorithm. Clustering is the most referenced application of unsupervised learning [14]. It consists of the distribution of the data between sub-sets by similarity (clusters) and allows discovering the intrinsic structure of the relationships between the data. For example, when we have several news articles and need to organize them to obtain those articles with similar topics.

3.7.2. Supervised Learning

In this case, both the training data and the test data are used. The first training set corresponds to the experience that contains significant information for the algorithm. The second set of tests allows to validate the acquired experience and predict the corresponding information. This can be understood as the process of transforming a particular input into a desired output, for this it must have gone through the learning of obtaining the output. For example, the detection of security events offered by a SIEM (Security and Event Management) system. This system has been trained by a wide set of events with different characteristics that later allows to identify if any new event

corresponds to a security incident and thus send the respective alarm. The green arrows show the model start-up process (Predictive Model), which was previously trained.

The model starts with the entry of a new data, in the example of the SIEM system this data could be the record of a service in a rush hour. The characteristics that the model requires for its evaluation are extracted from this register. Next, the model receives this vector and produces a label (Expected label) that corresponds to the prediction: Is the event a safety event or not? This process is repeated every time it receives a new vector of characteristics, whose label is unknown and it is expected to know.

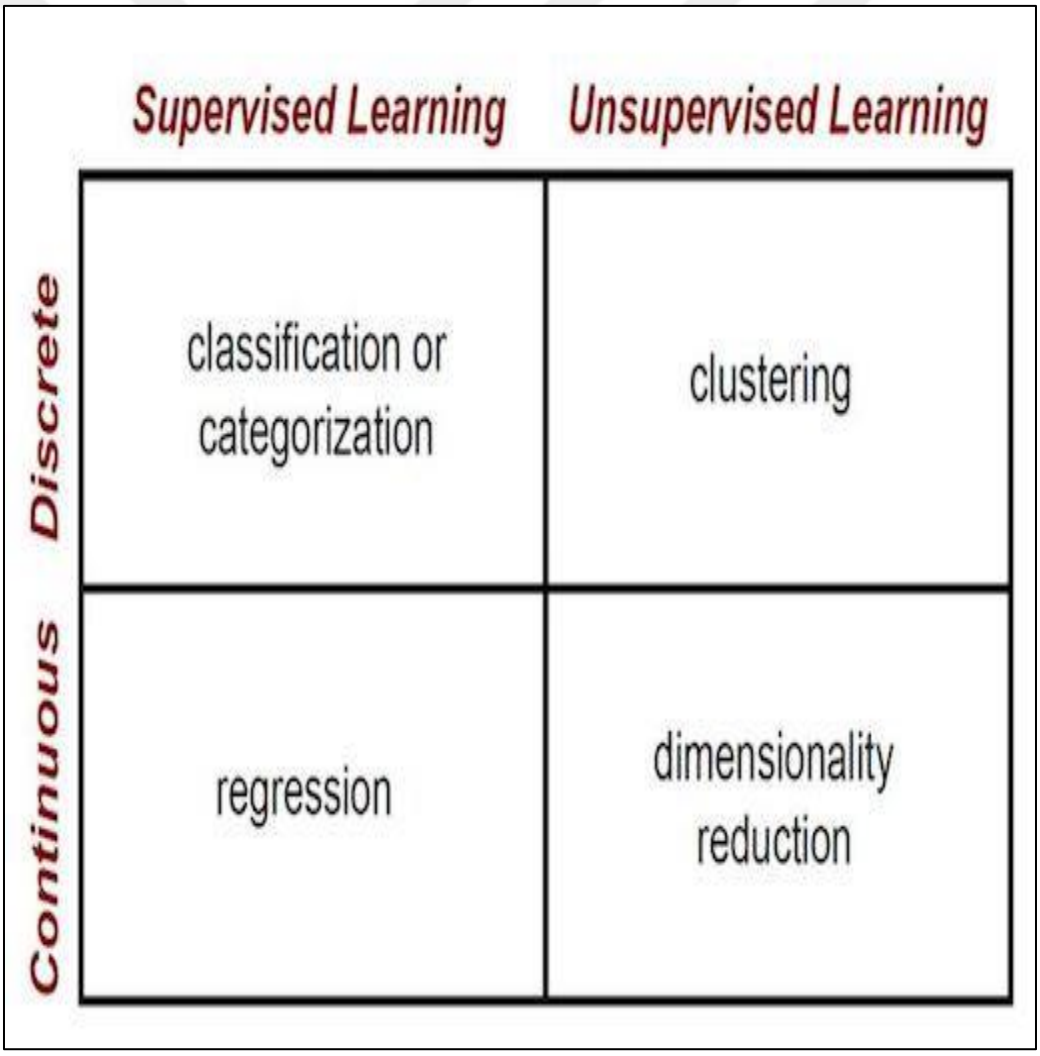

Figure 3.5: Classification of ML algorithms.

3.7.3. Application of Machine Learning

The set of Machine Learning algorithms is applied to numerous problems. For example, when publishing a photograph of people on social networks, the recognition of the characteristics of the faces is executed and then the name of each person is suggested. In the case of malware detection by antivirus systems, training is carried out using the collection of signatures or behavior patterns of viruses. Similarly, the mitigation of fraud in electronic commerce involves training the user's transactional activities, location, amount, periodicity and other purchase variables, which are verified by the model to alert a possible fraudulent transaction. Another application is the prediction of a woman's pregnancy based on her shopping habits, if she makes payments with coupons from magazines, if she buys for a small amount or if she stops buying some products, etc. In some applications the amount of data exceeds thousands, for example: sentiment analysis on soccer championship blogs. The algorithm will require each message with its respective label: {joy, disappointment, antipathy, neutral}. This task will require reading and tagging each message. For this, Crowdsourcing services are used, which consists of obtaining the work of people generally through the Internet to develop a task. It is necessary to clarify that both the obtaining of data, selection of characteristics and labeling of training data, make up the finest and most delicate tasks to perform to obtain a well-qualified prediction model.

It is also known as variable selection or attribute selection. It is the process of choosing the characteristics that are the most useful or relevant to solve a certain problem. If the characteristics are chosen so that there may be characteristics with similar values for different results then it does not matter how good the ML algorithm is.

Two types of feature selection algorithms are distinguished [16]:

i. Wrapper: The evaluation and selection of the attributes is performed by the algorithm based on the model's accuracy estimates. Combinations of the characteristics are made, selecting those that obtain the most accurate result for the model.

ii. Filter: Works separately from the learning algorithm and applies statistical measurements to assign a score to each characteristic. The Chi square test and the correlation of coefficients are some of the most widely used methods.

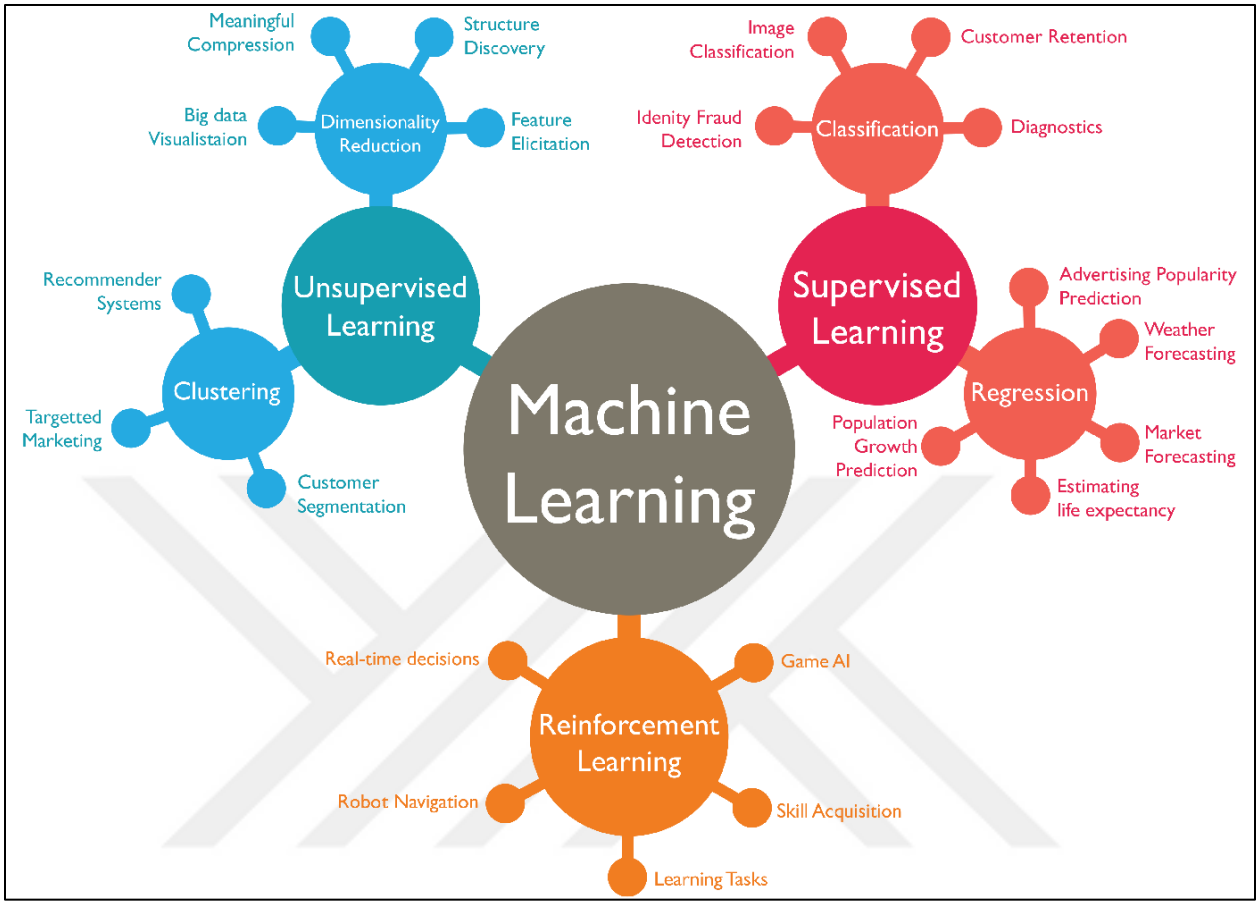

Figure 3.6: Applications of ML algorithms.

3.8 SCADA

SCADA systems are implemented, in most cases, in industrial environments in order to centralize the information from sensors, actuators and controllers and thus have remote control over an entire plant. For this to be possible, several elements are necessary, which are finally in charge of manipulating, transforming and transporting the information so that the processes can be executed automatically. Additionally, the systems are configured with different operating parameters defined by the personnel in charge [10]. Regarding the disciplines involved in industrial automation, the following can be mentioned: Automatic control, which is the discipline that allows the modeling of processes and their programming [11]. Industrial instrumentation, which is the area in charge of defining the correct selection of components, while data supervision is the branch in charge of information management and alarm systems [12] [13]. The hierarchy of processes is modeled taking as a reference the control pyramid, which will be discussed below.

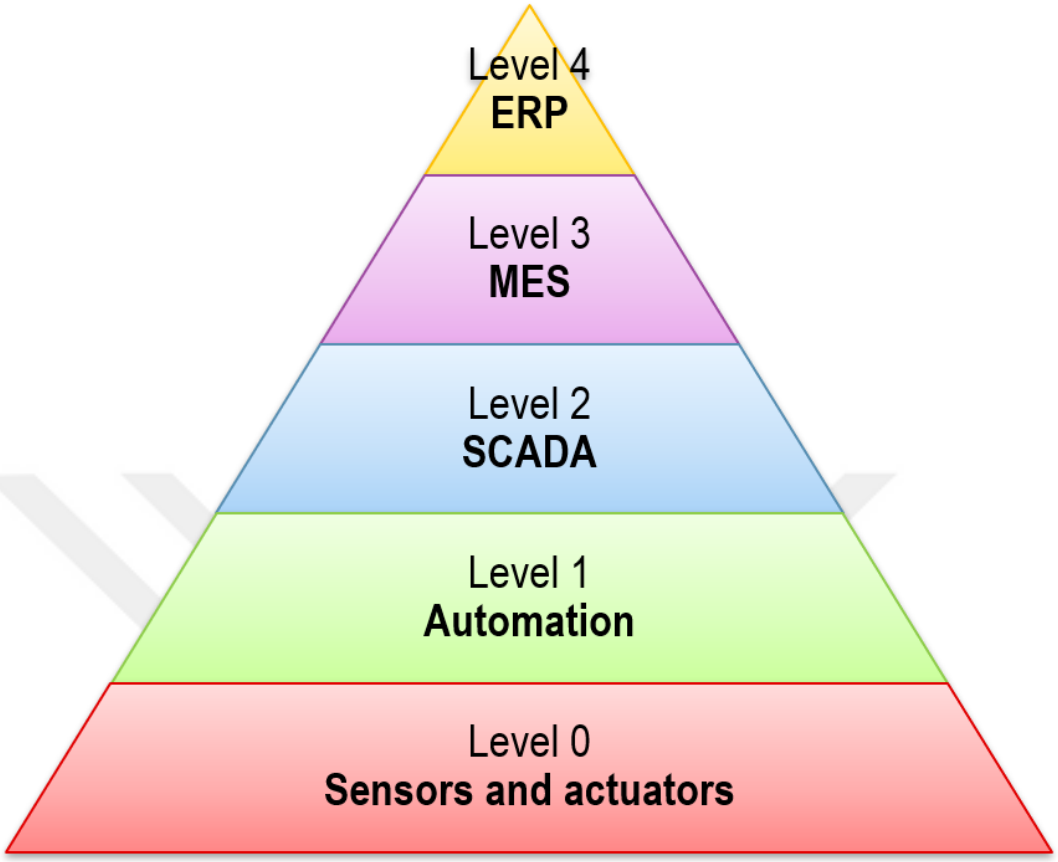

Figure 3.7: Hierarchy of SCADA.

The Control pyramid defines the hierarchical structure of remote management systems or SCADA [14] describe the different distributions that compose it, as well as the different control elements that are related in each one of them. As can be seen in the figures, the control and visualization systems are part of level 3, while the control devices are located in level 2. The system proposed in this work is installed between levels 2 and 3 alerting on any possible anomaly in the normal operation of the processes using the analysis of the different variables of the controllers involved. It should be noted that as the level increases, the volume of information grows dramatically and therefore the speed of response; An effective detection system must therefore be located in the lower parts so that it can promptly alert to a possible anomaly.

The basic scheme of a SCADA system is made up of the elements described; There, the connection diagram and the interaction between the different elements are shown. Each of the numerals that appear in the figure are described below:

a. Controllers (PLC): They are programmable devices that allow the automation of processes. The PLCs are in charge of managing the information coming from the sensors and at the same time executing the control actions on the actuating devices; In recent years they have been endowed with great versatility, which makes it possible to establish advanced communications based on standards such as: Ethernet, RS485, RS232, among others.

b. Communications: They are those cables and devices that allow interaction between the control devices and the servers. Depending on the needs or the technology used, the type of communication protocol and interface devices are selected: cables, routers, among others. In recent years, most of the automatisms come equipped with an Ethernet port; This standard facilitates the interaction between the automation stage and traditional information networks.

c. SCADA server: It is the system where all the information from the different controllers associated with a plant is centralized. It is a high capacity system that interacts with many devices in real time in such a way that client-type devices can have access to the different supervision variables and at the same time can send commands through the control variables.

d. Databases: Although they are normally embedded within the SCADA server, it is these that allow the storage of historical data of the different processes. Their role is crucial when it comes to analyzing the efficiency of the processes and reporting possible anomalies in their normal execution. and.

e. Human-Machine Interface (HMI): It could be said that they are the client devices in a SCADA network. Through these it is possible to interact with the different processes in a visual and user-friendly way. They are characterized by having animations that model the behavior of a number of control and observation variables, as well as the different alarm panels, in case of failure or malfunction of the processes.

f. Field Buses: These are networks of actuators and sensors that allow the controlling device to manage it, without the need to implement a large number of cables or establish point-to-point connections with each of them. In short, what is sought is to simplify the connections between the sensing, control and action devices; in such a way that it is much easier to detect possible failures

while the scalability of the processes is guaranteed. Compared with information networks, field buses are characterized by having smaller volumes of information and high response speeds [15].

g. Actuator Devices: They are those in charge of executing the control actions in the different processes when the master device so determines. Within the category of actuating devices, we can find: Motors, valves, pistons, contactors, etc.

h. Sensor Devices: They are the devices in charge of converting any type of physical magnitude of a process into an equivalent voltage or current signal. Through them, the controllers obtain information on the decision-making process. Given the heterogeneity of SCADA systems and the diversity of protocols and devices that can be used in their implementation, the usefulness of SCADA systems based on the OPC standard is described below. This standard constitutes a key piece in the approach and development of the work presented here.

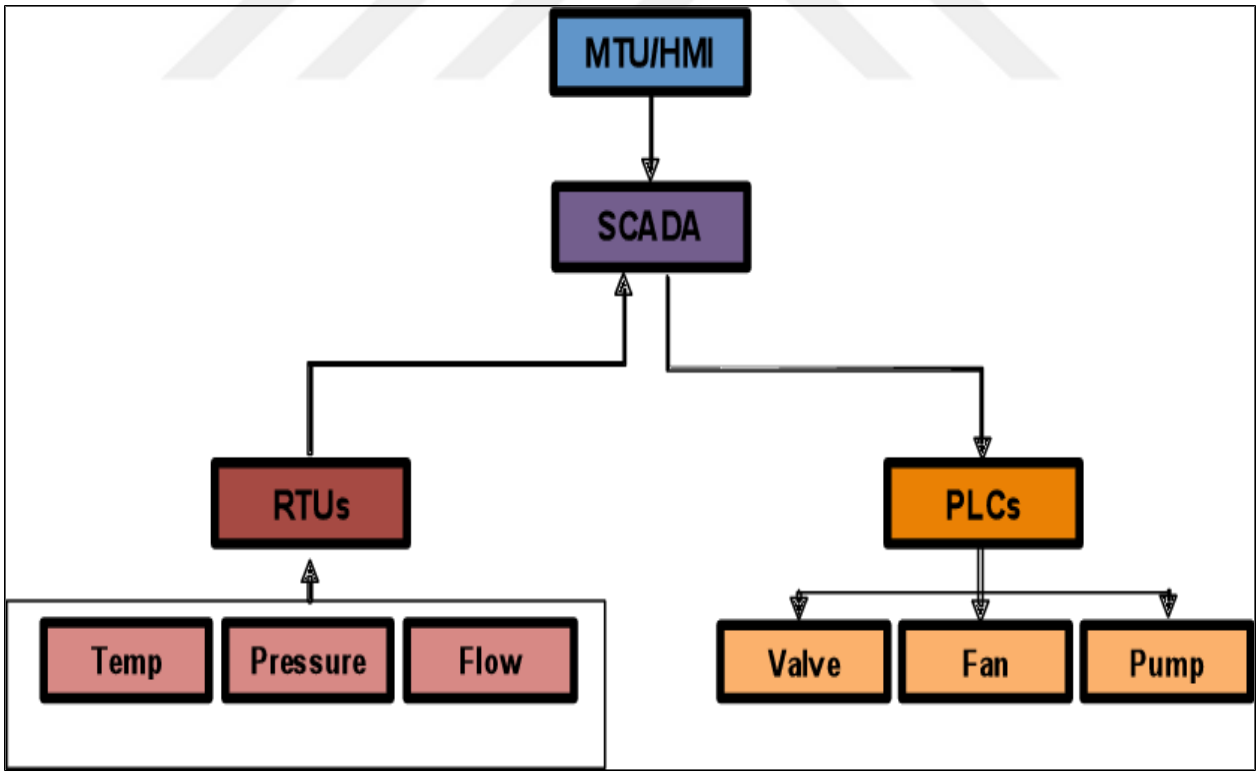

Figure 3.8: The basic scheme of a SCADA system.

4. PROPOSED METHOD

4.1 SYSTEM DESIGN

The MATLAB programming software, as already mentioned in previous sections, consists of a graphical programming environment that is composed of 2 interfaces called: Control Panel and Block Diagram as shown in figure 4.1 In the block diagram interface, Programming locates all the control and visualization elements, such as: buttons, sensors, knobs, graphs among others, while the interconnection of the modules and all the programming associated with them is carried out in the block diagram interface.

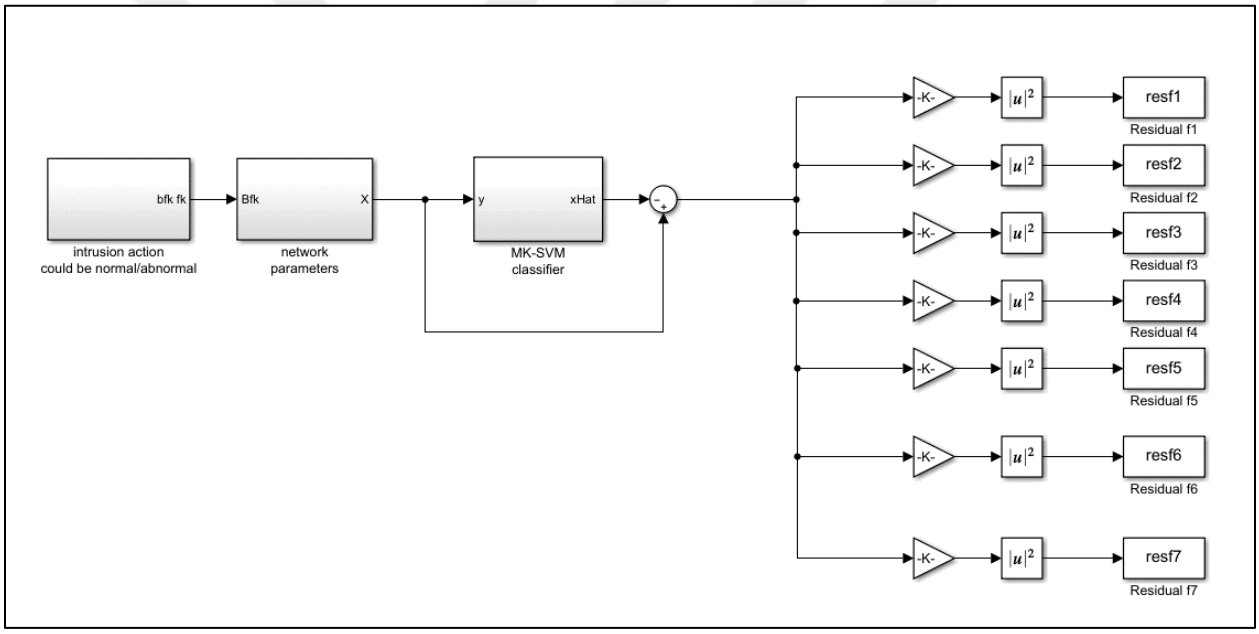

Figure 4.1: MATLAB Simulink simulation of SCADA system.

Undoubtedly, this powerful graphical programming environment has functionalities that go beyond virtual instrumentation, allowing all the necessary components for the development of the proposed intrusion detector to be implemented in a single interface. In particular, through the LabVIEW software it is possible to configure a client interface that links to the OPC server, allowing, through this functionality, not only the detector interface but also to implement an intrusive application whose objective is to try alter the normal operation of a given process SCADA systems, like traditional computer systems, have become the target of all kinds of cyber-

attacks that can seriously compromise the critical infrastructure of a country, compromise the productivity of many companies or at worst, cause loss of human life. In most cases, as will be seen in the next section, the traditional attacks to which a traditional computer system is subjected apply to SCADA systems; However, the repercussions caused by the attack on a control system in an industrial plant have more media effects on the performance of production systems.

4.2 IMPLEMENTING ATTACKS ON SCADA

Once some generalities of SCADA systems and the possible types of attacks that could be victims have been described, it is convenient to illustrate 3 types of scenarios that the solution that will be described in this work can detect. In figure 4.2, the location of the attacker is shown, while the possible consequences that such an attack could have are described. The first scenario, an intruder impersonates the identity of the server, so that he can tamper with the parameters of the drivers or even alter the true value of the observed variables before they reach the SCADA server. This type of attack is known in as "Man in the middle" and can be potentially harmful for the correct operation of a SCADA system. In the second scenario, an intruder takes control of a Client platform and through it manipulates the control parameters improperly; In addition, the intruder can filter information from the observed variables. Compared to the first scenario, the possibility of an attack like this being carried out is much higher since most security tools prioritize the server. In the third scenario, a “malware” infects the client or server platform and from there it improperly manipulates the control or observation parameters. This scenario is one of the most dangerous, since the attack is carried out silently and it is likely that traditional mechanisms will not be able to detect it. If it is also considered that many industries in Latin America do not have the latest software updates for their respective computing platforms, the probability of an infection of this type is extremely high. The system described in this work has the ability to detect attacks in the 3 types of scenarios shown; However, its main strength is in the detection of attacks type "Malware" where the variation of parameters can become imperceptible for the traditional alert systems of a SCADA system.

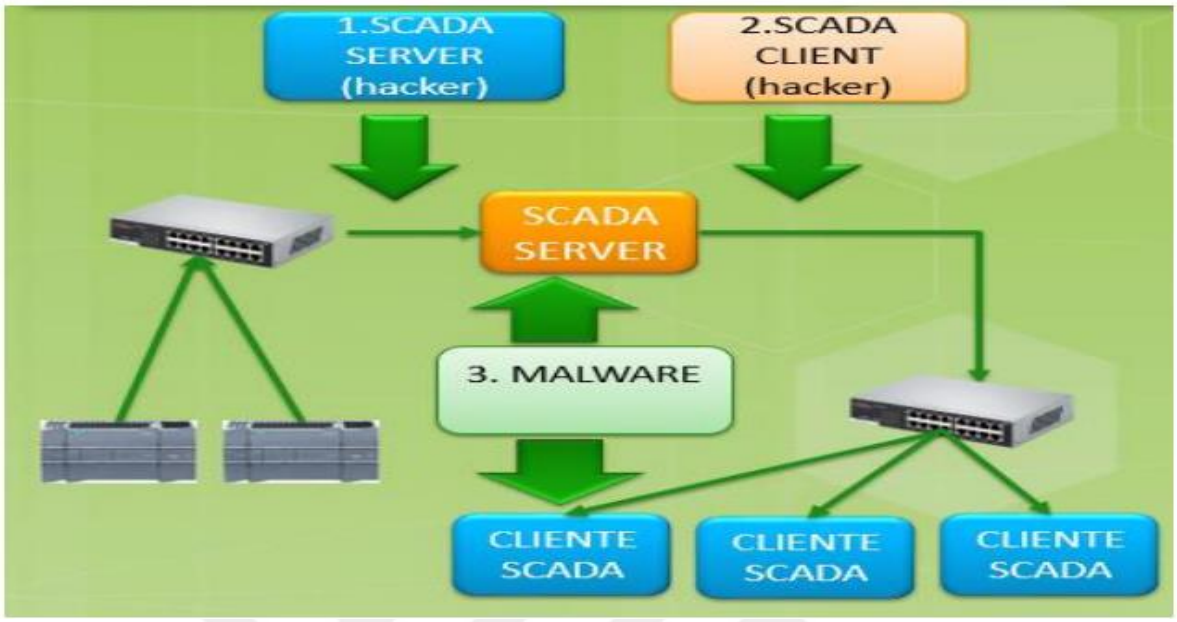

Figure 4.2: Man in the middle attack on SCADA.

4.3 DETECTING ATTACKS ON SCADA USING SVM

Intrusion detection systems or IDS are systems in charge of analyzing the traffic, behavior of a network or device in order to detect anomalous patterns, while reporting possible incidents. IDS can be classified into 2 types depending on where the detection will take place.

The N-IDS are systems that allow to examine the anomalous behavior of a network or at least part of it, while the H-IDS allow to evaluate unwanted patterns within an end device or Host [26]. For the specific case of SCADA systems, both approaches could be possible; where the N-IDS allow the monitoring of each of the communication protocols between the control units while the H-IDS allow the evaluation of the process variables within the controller. Depending on your configuration and the type of systems you are looking to monitor.

Classic IDS systems use various techniques to detect potential cyberattacks. The first technique for intrusion detection is to detect corrupted packets or in other words, packets with information that is harmful to network components. For the application of this technique, the system must

recognize the type of protocol used and thus verify that each of the fields is within its normal ranges. This technique could be quite useful in SCADA systems where the number of implemented protocols is not very large, thus allowing none of the system variables to be outside their normal range. The second technique for intrusion detection consists of identifying attack patterns which are characterized by non-regular behavior. In this case, the IDS analyze the behavior of a specific network by monitoring the route of packets and their corresponding sender and receiver. If eventually a package or transaction does not behave in the desired way, this is classified as a possible attack until the respective review. If the registered packet does correspond to an attack, the IDS system is trained in such a way that it can identify similar attacks. In the particular case of SCADA systems, this approach can be given from 2 independent fronts or a combination of them. In the first case, the IDS system only analyzes the behavior of the variables of a system and based on them it could detect a possible threat. In the second, for its part, the network packets in charge of controlling the process are analyzed and it is verified that these are within the desired ranges. The combination of both approaches is more effective since it can discern if the anomalous behavior of the plant is due to a malfunction of the same or a cybernetic attack [28], where the characteristics that an intrusion detection system must have within an industrial control network are defined. According to this document, the system must, first of all, identify very well the type of protocol implemented in the SCADA network so that it can comply with a second very important characteristic, which is packet penetration. At this stage, the system is capable of detecting corrupt information that puts the monitored plant at risk. The third characteristic that must be fulfilled is the identification of patterns, so that the IDS system has the ability to adapt to the behavior of the SCADA network and detect when abnormal behavior occurs and as the last characteristic it must be able to identify critical states in the plant as a product of malicious packages. So far, an introduction has been made to each of the topics of interest related to SCADA systems. In the next section, it will be described what the automatic learning methods consist of and then address each of the works carried out in the world of IDS-SCADA.

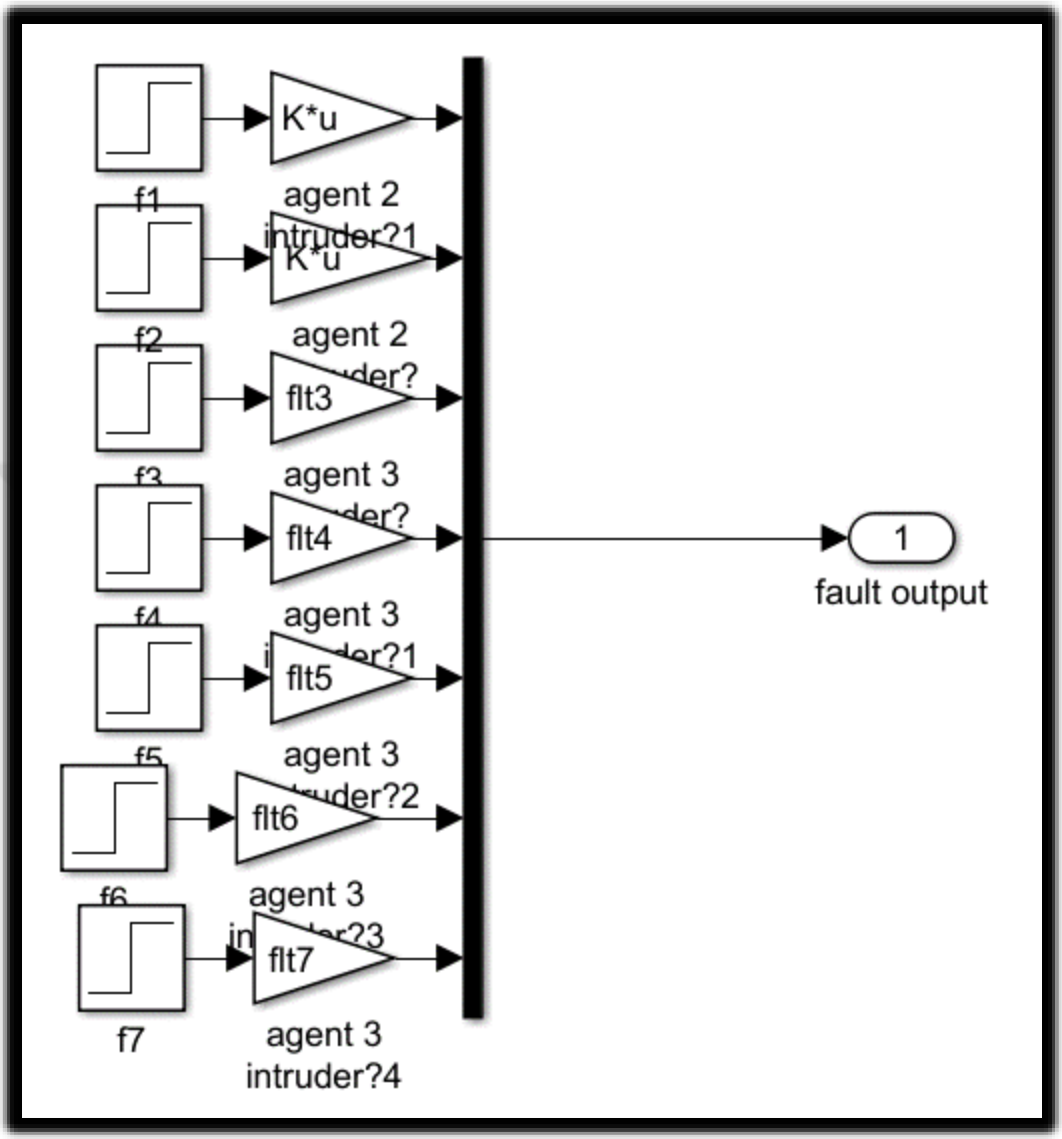

Figure 4.3: Agents or Intruders detector in the proposed SCADA.

Vector support machines of the "One Class" type represent a special case of biclass classification, where the classification algorithm is only trained with one type of data, allowing the subsequent detection of atypical patterns that do not conform to the previously established model. By default, vector support machines are trained by 2 types of samples where each of them is assigned a label for later identification. Typically, a 1 or -1 is assigned to differentiate the classes; However, in the SVM the training is only carried out with a type of data that could well be identified with a 1. Already in the classification stage, the behavior of both methodologies is practically the same,

with the difference that for the SVM Traditional labels thrown in the classification correspond to previously defined classes, while for SVM a 1 represents that the new data conforms to the model provided or -1 that said data has a different behavior than expected. The fact that SVM can be trained with a single type of data, allows it to be widely used in the detection of outliers or "Outliers", such as, for example, in the detection of possible tampering with the variables of a SCADA system [3]. As in traditional SVMs, the algorithm calculates the decision boundary based on some data called support vectors. In the case of SVM, the boundary is calculated using only one data type as shown in figure 2. The SVM algorithm first maps the input data on a higher dimensional scale through a Kernel function and then iteratively Find the marginal separation plane that allows the best separation of the data with respect to the origin. In this way, the hyperplane or decision boundary is described by the classification.

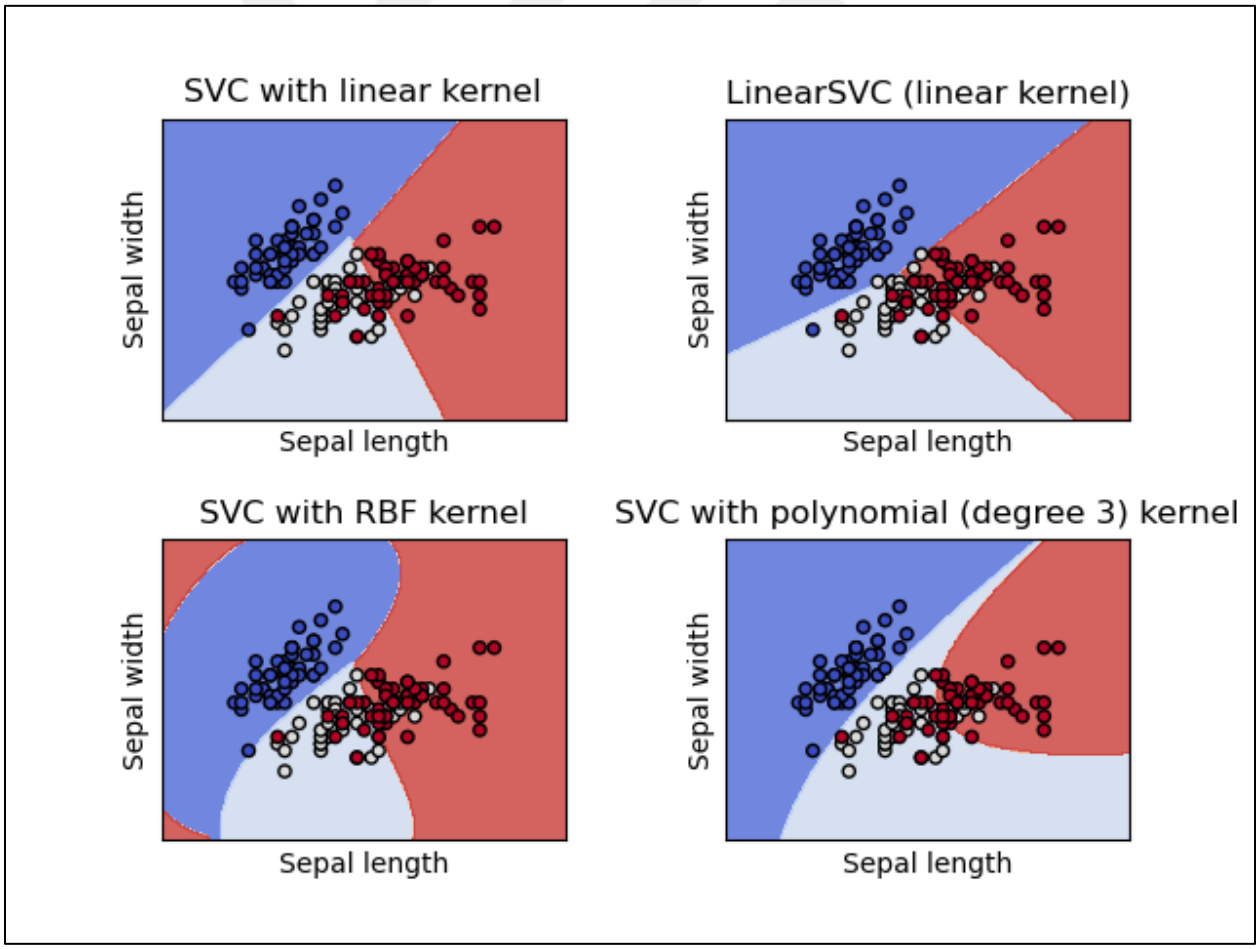

Figure 4.4: SVM Classification with multiple kernels.

The SVM method, being a variation of the original biclass SVM method, only requires one type of sample for training; however, for real-time analysis it is quite effective since it requires less execution time. The improvement to the initial system proposed in [48] is described in [49] by adding the Kmeans clustering algorithm to the initial IDS. K-medias groups the network packets of the SCADA system according to their common characteristics, characterizing the corrupt packets among them. The correlation of the results obtained by the clustering algorithms is ultimately what allowed the author to improve the efficiency of the proposed IDS. According to the published results, this approach reaches almost 100% efficiency; However, if it is considered that only 2 training characteristics were taken into account (packet size and update rate), it is not possible to give absolute credibility to this result based on the few descriptors.

Figure 4.5: Sample of the detection result.

The table below compares the results of our work and previous works done in the same subject.

Table 4.1: Comparison of results on SCADA attacks

Method	**Algorithm**	**Attack**	**Scope of improvement**	**Detection**
proposed Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS)	Machine learning classification of abnormal activity	Man, in the middle, packet corruption.	Desirable conditions are specified for an IDS for SCADA systems.	Good - Excellent
Snort IDS for SCADA networks	Snort is used as IDS in a network made up of virtual and real units	Not specified	Honeypot implementation to identify other possible attacks.	Good
A Multidimension al Critical State Analysis for Detecting Intrusions in SCADA Systems	Intrusion detection in SCADA systems by monitoring traffic between controllers under the PROFINT standard	Package corruption and external interventions	Testing against the standard and the aforementioned attacks.	Fair
RULE-BASED INTRUSION DETECTION SYSTEM FOR SCADA NETWORKS	Signature based. -model based for intrusion detection	DDOS	It is not specified in the article, but in 2014 a new article was written addressing a new technique called multi-attribute IDS for SCADA systems.	Good
Network Anomaly Detection for m-connected SCADA Networks	network-based pattern reference. The proposed IDS generate dynamic rules to detect a possible attack.	Package corruption	The article has a descriptive approach; therefore, it only addresses the possible efficacy of the described technique.	fair

Given the results obtained about the detections carried out, as shown in above it is observed that, this time, only two machine learning techniques - Note, also, that the two techniques that achieve success in the detection of this attack are not those that will present the best results in the first stage of two experiments based on data, showing that, in the face of unfamiliar network traffic flows, Decision Tree and Linear Support Vectors Machine has a higher generalization capacity when compared to AdaBoost and Random Forests. Therefore, it is understood that the objective is to detect attacks, it is very efficient and effective to opt for the use of Linear Support Vectors Machine for To this end, considering that the techniques will not barely detect all the attacks, as well as, they will present an average classification time significantly lower than the AdaBoost in Random Forests, or that it can be a decisive factor for the escort of a learning technique. of machine for a high-speed network environment. It should be noted, furthermore, that a strong point of the approach of detection of intrusion based on anomaly and that, differently from the approach based on is still extremely important. for the maintenance and protection of computer networks. In a large majority of cases where IDSs based on assumptions are correctly configured, taking into account the particularities of the network that is planned to be monitored, they will be less likely to be subject to false alerts than those addressed by anomalies based on anomalies. However, nothing prevents hybrid IDSs from being implanted in computer network projects. In a hybrid configuration of Snort 3, for example, an approach based on assumption could be used for the easily mitigated, whereas, on the other hand, an approach based on anomaly could be concentrated in the containment of most complex, dynamic, where, in this setting, a classification model could be more appropriate, as observed in our experiments conducted over the course of this stage.

Finally, the figure below showing the accuracy of algorithms in attack detection.

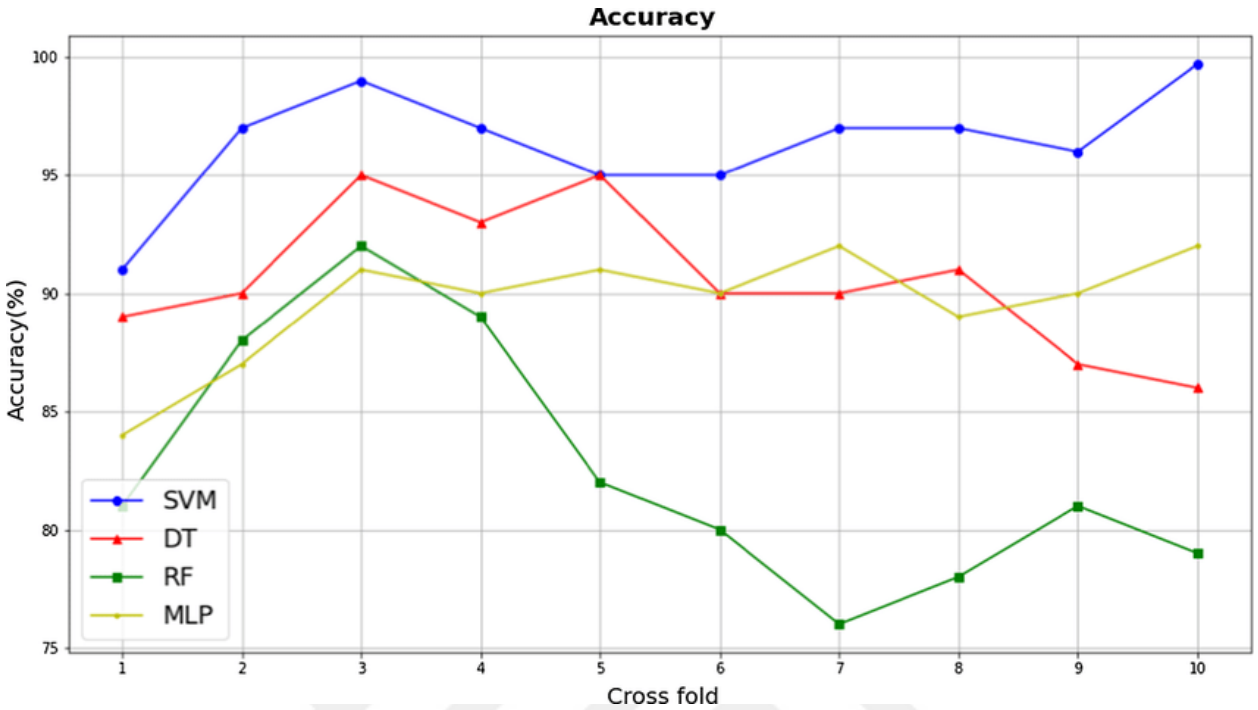

Figure 4.6: Accuracy of algorithms in attack detection.

5. CONCLUSION

In this thesis, we were able to explore the possibilities offered by intrusion detection in industrial systems. The characteristics of the latter mean that this security measure can give excellent results while the non-intrusive nature of IDS facilitates their deployment and use. As a proof of concept, a complete preprocessor for the Modbus protocol has been developed within the IDS Suricata. Its imminent integration into the project will make it accessible to as many people as possible for experimentation or even operational use. The performance tests carried out show the feasibility of detection probes with inexpensive equipment. However, before arriving at a fully operational product, much work remains to be done. To facilitate this task, it seems essential to develop tools to assist experts from the design phase of the system and throughout its existence. If the mapping is sufficiently precise, such a tool can also automatically generate the rules for detecting IDS. In this evolutionary stage in which the technology is currently, the ability of a device to connect to the Internet is seen as an indispensable prerequisite for it to be considered relevant in the context of technological innovations. As a result of this phenomenon - which says a lot about the way technology touches people's daily lives - it is clear that computer networks are in a state of constant renewal, not only to support the adoption of new devices but, also, to ensure the security and integrity of all components that make up their architectures. The growth of computer networks, however, goes against their security, since, as they become more heterogeneous, their attack surfaces increase, that is, their penetration amplitudes. In the area of intrusion detection, known attacks are usually contained with some effectiveness through signature-based approaches, employed in the vast majority of Network Intrusion Detection Systems, such as MATLAB machine learning library for SCADA. On the other hand, unknown attacks increasingly relevant given the frequency with which they emerge over the years are contained through anomaly-based approaches, usually making use of machine learning techniques. Considering the challenge of dealing with unknown attacks and, consequently, adapting existing tools to contain these attacks, this work proposed the replacement of the signature-based detection scheme of the latest version of (MATLAB machine learning library for SCADA) by an anomaly-based detection, to enable real-time classification of network traffic flows through the use of machine learning techniques, which until now, are non-existent in IDS. For that, a plugin was developed for the task in question, called ML classifiers.

REFERENCES

[1] Nembhard, A. D., Sinha, J. K., Elbhbah, K., Pinkerton A. J.: *Fault diagnosis of rotating machines using vibration and bearing temperature measurements*. Diagnostyka, vol. 14 (3), 45-52 (2013).

[2] Tabaszewski, M.: *Optimization of a nearest neighbors' classifier for diagnosis of condition of rolling bearings*. Diagnostyka, vol. 15 (1), 37-42 (2014).

[3] Astolfi, D., Castellani. F., Terzi, L.: *Fault prevention and diagnosis through SCADA temperature data analysis of an onshore wind farm*. Diagnostyka, vol. 15, 71-78 (2014).

[4] Chehri, A., Jeon, G.: The Industrial Internet of Things: *Examining How the IoT Will Improve the Predictive Maintenance*. Lecture Notes of the Institute for Computer Sciences, Smart Innovation Systems and Technologies, Springer, Heidelberg (2019).

[5] Chehri, A., Jeon, G.: *Routing Protocol in the Industrial Internet of Things for Smart Factory Monitoring*. Lecture Notes of the Institute for Computer Sciences, Smart Innovation Systems and Technologies, Springer (2019).

[6] Jeon, G., Awais, A., Chehri, A., Cuomo, S.: *Special issue on video and imaging systems for critical engineering applications.* Multimedia Tools and Applications, Springer (2020).

[7] Jeon, G., Chehri, A, Cuomo. S, Din. S, Jabbar, S.: *Special Issue on Real-time Behavioral Monitoring in IoT Applications using Big Data Analytics*. Concurrency and Computation: Practice and Experience, John Wiley and Sons (2019).

[8] Saufi, S. R., Ahmad, Z. A. B., Leong. M. S., Lim. M. H: *Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis*: A Review. IEEE Access, vol. 7, 122644-122662 (2019).

[9] Zhang, S., Zhang, S, Wang. B, Habetler, T. G.: *Machine learning and deep learning algorithms for bearing fault diagnostics*—A comprehensive review (2019).

[10] Jaafar, A.: *Vibration Analysis and Diagnostics Guide*. College of Engineering, University of Basra (2012).

[11] Yang, W., Court, R., Jiang. S.: *Wind turbine condition monitoring by the approach of SCADA data analysis*. Renew. Energy 53 (2013).

[12] Lee, J., Qiu, H., Yu, G., Lin. L.: *Bearing Data Set. NASA Ames Prognostics Data Repository*, NASA Ames Research Center, Moffett Field, University of Cincinnati (2004).

[13] Saber, M., Saadane, R., Aroussi. H., Chehri, A.: *An Optimized Spectrum Sensing Implementation based on SVM, KNN and TREE Algorithms*. IEEE 15th International Conference on Signal Image Technology & Internet Based Systems, Sorrento (NA), Italy (2019).

[14] Boser, B., Guyon, I., Vapnik, V.: *A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, Pittsburgh, Pennsylvania, United States, 144–152 (1992).

[15] Yamada, Y., Suzuki, E., Yokoi, H., Takabayashi, K.: *Decision-tree induction from timeseries data based on a standard-example split test*. Machine Learning, Proceedings of the Twentieth International Conference, Washington, DC, USA (2003).

[16] Kilian, Q., Weinberger, J., Blitzer, J., Saul, L.K.: *Distance metric learning for large margin nearest neighbor classification*. In Advances in neural information processing systems, 1473–1480 (2006).

[17] Friedman, N., Geiger, D., Goldszmidt, M.: *Bayesian network classifiers*. Machine Learning. vol. 29, 131-163 (1997).

[18] Igure, V., Laughter, S., Williams, R.: *Security issues in SCADA networks*. Computers & Security 25(7), 498–506 (2006)

[19] Slay, J., Miller, M.: *Lessons learned from the maroochy water breach*. International Federation for Information Processing 253, 73 (2008)

[20] Beckner, W.D.: NRC Information Notice 2003-14: *Potential Vulnerability of Plant Computer Network to Worm Infection* (2003)

[21] Luiijf, E.: *SCADA Security Good Practices for the Drinking Water Sector*. Technical Report - tno.nl (2008)

[22] Stouffer, K., Falco, J., Kent, K.: *Guide to supervisory control and data acquisition (SCADA) and industrial control systems security* (2006)

[23] Naess, E., Frincke, D., McKinnon, A., Bakken, D.: *Configurable Middleware Level Intrusion Detection for Embedded Systems*. In: 25th IEEE International Conference on Distributed Computing Systems Workshops, pp. 144–151 (2005)

[24] Valdes, A., Cheung, S.: *Intrusion Monitoring in Process Control Systems*. In: Proceedings of the Forty-Second Hawaii International Conference on System Sciences, p. 17 (2009)

[25] C´ardenas, A., Amin, S., Sastry, S.: *Research challenges for the security of control systems. In: Proceedings of 3rd USENIX workshop on Hot Topics in Security (HotSec)*, San Jose, CA, USA (2008)

[26] Rrushi, J., Kang, K.d.: *Detecting Anomalies in Process Control Networks*. In: Critical Infrastructure Protection III: Third IFIP WG 11.

[27] International Conference, Hanover, New Hampshire, USA, pp. 151–165. Springer, Heidelberg (2009)

[28] Kannan, J., Jung, J., Paxson, V., Koksal, C.: *Semi-automated discovery of application session structure.* In: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, p. 132. ACM, New York (2006)

[29] M.S. Alam and S.T. Vuong. *Random forest classification for detecting android malware*. In Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (I Things/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, pages 663–669, Aug 2013. doi: 10.1109/GreenComiThings-CPSCom.2013.122.

[30] P.P.K. Chan and Wen-Kai Song. *Static detection of android malware by using permissions and api calls. In Machine Learning and Cybernetics (ICMLC)*, 2014 International Conference on, volume 1, pages 82–87, July 2014. doi: 10.1109/ICMLC.2014.7009096.

[31] Hsin-Yu Chuang and Sheng-De Wang. *Machine learning based hybrid behavior models for android malware analysis*. In Software Quality, Reliability and Security (QRS), 2015 IEEE International Conference on, pages 201–206, Aug 2015. doi: 10.1109/QRS.2015.37.

[32] M. Fazeen and R. Dantu. Another free app: *Does it have the right intentions? In Privacy*, Security and Trust (PST), 2014 Twelfth Annual International Conference on, pages 282–289, July 2014. doi: 10.1109/PST.2014.6890950.

[33] S. Feldman, D. Stadther, and Bing Wang. Manilyzer: Automated android malware detection through manifest analysis. In Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on, pages 767–772, Oct 2014. doi: 10.1109/MASS.2014.65.

[34] M. Ghorbanzadeh, Yang Chen, Zhongmin Ma, T.C. Clancy, and R. McGwier. *A neural network approach to category validation of android applications. In Computing, Networking and Communications (ICNC),* 2013 International Conference on, pages 740–744, Jan 2013. doi: 10.1109/ICCNC.2013.6504180.

[35] W. Glodek and R. Harang. *Rapid permissions-based detection and analysis of mobile malware using random decision forests*. In Military Communications Conference, MILCOM 2013 - 2013 IEEE, pages 980–985, Nov 2013. doi: 10.1109/MILCOM.2013.170.

[36] Hyo-Sik Ham and Mi-Jung Choi. *Analysis of android malware detection performance using machine learning classifiers*. In ICT Convergence (ICTC), 2013 International Conference on, pages 490–495, Oct 2013. doi: 10.1109/ICTC.2013.6675404.

[37] Wan-Chen Hsieh, Chuan-Chi Wu, and Yung-Wei Kao. *A study of android malware detection technology evolution*. Balaji Baskaran and Anca Ralescu MAICS 2016 pp. 15–23 21 In Security Technology (ICCST), 2015 International Carnahan Conference on, pages 135–140, Sept 2015. doi: 10.1109/CCST.2015.7389671.

[38] Ideses and A. Neuberger. *Adware detection and privacy control in mobile devices*. In Electrical Electronics Engineers in Israel (IEEEI), 2014 IEEE 28th Convention of, pages 1–5, Dec 2014. doi: 10.1109/EEEI.2014.7005849.

[39] F. Idrees and M. Rajarajan. *Investigating the android intents and permissions for malware detection*. In Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on, pages 354– 358, Oct 2014. doi: 10.1109/WiMOB.2014.6962194.

[40] Q. Jerome, K. Allix, R. State, and T. Engel. *Using opcode-sequences to detect malicious android applications*. In Communications (ICC), 2014 IEEE International Conference on, pages 914–919, June 2014. doi: 10.1109/ICC.2014.6883436.

[41] Hwan-Hee Kim and Mi-Jung Choi. *Linux kernel-based feature selection for android malware detection*. In Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific, pages 1–4, Sept 2014. doi: 10.1109/APNOMS.2014.6996540.

[42] H. Kurniawan, Y. Rosmansyah, and B. Dabarsyah. *Android anomaly detection system using machine learning classification*. In Electrical Engineering and Informatics (ICEEI), 2015 International Conference on, pages 288–293, Aug 2015. doi: 10.1109/ICEEI.2015.7352512.

[43] M. Lindorfer, M. Neugschwandtner, and C. Platzer. Marvin: *Efficient and comprehensive mobile app classification through static and dynamic analysis*. In Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual, volume 2, pages 422–433, July 2015. doi: 10.1109/COMPSAC.2015.103.

[44] Wen Liu. *Multiple classifier system based android malware detection*. In Machine Learning and Cybernetics (ICMLC), 2013 International Conference on, volume 01, pages 57–62, July 2013. doi: 10.1109/ICMLC.2013.6890444.

[45] Xing Liu and Jiqiang Liu. *A two-layered permission-based android malware detection scheme*. In Mobile Cloud Computing, Services, and Engineering (Mobile Cloud), 2014 2nd IEEE International Conference on, pages 142–148, April 2014. doi: 10.1109/MobileCloud.2014.22.

[46] Yu Lu, Pan Zulie, Liu Jingju, and Shen Yi. *Android malware detection technology based on improved Bayesian classification*. In Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2013 Third International Conference on, pages 1338–1341, Sept 2013. doi: 10.1109/IMCCC.2013.297.

[47] M.Z. Mas'ud, S. Sahib, M.F. Abdollah, S.R. Selamat, and R. Yusof. *Analysis of features selection and machine learning classifier in android malware detection*. In Information Science and Applications (ICISA), 2014 International Conference on, pages 1–5, May 2014. doi: 10.1109/ICISA.2014.6847364.

[48] Munoz, I. Martin, A. Guzman, and J.A. Hernandez. *Android malware detection from google play meta-data: Selection of important features*. In Communications and Network Security (CNS), 2015 IEEE Conference on, pages 701–702, Sept 2015. doi: 10.1109/CNS.2015.7346893.

[49] D.V. Ng and J.-I.G. Hwang. *Android malware detection using the dendritic cell algorithm. In Machine Learning and Cybernetics (ICMLC),* 2014 International Conference on, volume 1, pages 257–262, July 2014. doi: 10.1109/ICMLC.2014.7009126.

[50] U. Pehlivan, N. Baltaci, C. Acarturk, and N. Baykal. *The analysis of feature selection methods and classification algorithms in permission based android malware detection*. In Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium on, pages 1–8, Dec 2014. doi: 10.1109/CICYBS.2014.7013371.

[51] N. Peiravian and Xingquan Zhu. *Machine learning for android malware detection using permission and api calls.* In Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on, pages 300–305, Nov 2013. doi: 10.1109/ICTAI.2013.53.

[52] R. Raveendranath, V. Rajamani, A.J. Babu, and S.K. Datta. *Android malware attacks and countermeasures: Current and future directions*. In Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on, pages 137–143, July 2014. doi: 10.1109/ICCICCT.2014.6992944.

[53] RiskIQ. Android malware attacks and countermeasures: *Current and future directions*. June 2014. J. Sahs and L. Khan. A machine learning approach to android malware detection. In

Intelligence and Security Informatics Conference (EISIC), 2012 European, pages 141–147, Aug 2012. doi: 10.1109/EISIC.2012.34.

[54] A.A.A. Samra, Kangbin Yim, and O.A. Ghanem. *Analysis of clustering technique in android malware detection*. In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on, pages 729–733, July 2013. doi: 10.1109/IMIS.2013.111.

[55] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, and P.G. Bringas. *On the automatic categorization of android applications*. In Consumer Communications and Networking Conference (CCNC), 2012 IEEE, pages 149–153, Jan 2012. doi: 10.1109/CCNC.2012.6181075.

[56] Shabtai. *Malware detection on mobile devices*. In Mobile Data Management (MDM*),* 2010 Eleventh International Conference on, pages 289–290, May 2010. doi: 10.1109/MDM.2010.28.

[57] Shabtai, Y. Fledel, and Y. Elovici. *Automated static code analysis for classifying android applications using machine learning.* In Computational Intelligence and Security (CIS), 2010 International Conference on, pages 329–333, Dec 2010. doi: 10.1109/CIS.2010.77.

[58] L. Tenenboim-Chekina, O. Barad, A. Shabtai, D. Mimran, L. Rokach, B. Shapira, and Y. Elovici. *Detecting application update attack on mobile devices through network feature.* In Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, pages 91–92, April 2013. doi: 10.1109/INFCOMW.2013.6970755.

[59] Balaji Baskaran and Anca Ralescu MAICS 2016 pp. 15–23 22 Te-En Wei, Ching-Hao Mao, A.B. Jeng, Hahn-Ming Lee, Horng-Tzer Wang, and Dong-Jie Wu. *Android malware detection via a latent network behavior analysis*. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, pages 1251–1258, June 2012. doi: 10.1109/TrustCom.2012.91.

[60] Yu Wei, Hanlin Zhang, Linqiang Ge, and R. Hardy. *On behavior-based detection of malware on android platform.* In Global Communications Conference (GLOBECOM), 2013 IEEE, pages 814–819, Dec 2013. doi: 10.1109/GLOCOM.2013.6831173.

[61] Westyarian, Y. Rosmansyah, and B. Dabarsyah. *Malware detection on android smartphones using api class and machine learning*. In Electrical Engineering and Informatics (ICEEI), 2015 International Conference on, pages 294–297, Aug 2015. doi: 10.1109/ICEEI.2015.7352513.

[62] Zhao Xiaoyan, Fang Juan, and Wang Xiujuan. *Android malware detection based on permissions.* In Information and Communications Technologies (ICT 2014), 2014 International Conference on, pages 1–5, May 2014. doi: 10.1049/cp.2014.0605.

[63] J. Xu, Y. Yu, Z. Chen, B. Cao, W. Dong, Y. Guo, and J. Cao. Mobsafe: *cloud computing based forensic analysis for massive mobile applications using data mining*. Tsinghua Science and Technology, 18(4):418–427, August 2013. doi: 10.1109/TST.2013.6574680.

[64] S.Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik. *A new android malware detection approach using Bayesian classification*. In Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, pages 121–128, March 2013. doi: 10.1109/AINA.2013.88.

[65] S.Y. Yerima, S. Sezer, and G. McWilliams. *Analysis of Bayesian classification-based approaches for android malware detection*. Information Security, IET, 8(1):25–36, Jan 2014a. ISSN 1751-8709. doi: 10.1049/iet-ifs.2013.0095.

[66] S.Y. Yerima, S. Sezer, and I. Muttik. *Android malware detection using parallel machine learning classifiers*. In Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on, pages 37–42, Sept 2014b. doi: 10.1109/NGMAST.2014.23.

[67] Alazab, Mamoun, Sitalakshmi Venkatraman, Paul Watters, and Moutaz Alazab. 2011. *Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures*. Proceedings of the 9-th Australasian Data Mining Conference, 171-181.

[68] Aliyev, Vusal. 2010. *Using honeypots to study skill level of attackers based on the exploited vulnerabilities in the network*. Chalmers University of Technology. Aquino, Maharlito. 2014. Fake BACS Remittance Emails Delivers Dridex Malware.

[69] [Accessed 15 February 2017]. Aziz, Naman. 2014. *Cybergate RAT Tutorial for Beginners*. WWW document. Available at: http://rattut.blogspot.fi/.

[70] Baldangombo, Usukhbayar, Nyamjav Jambaljav, and Shi-Jinn Horng. 2013. *A Static Malware Detection System Using Data Mining Methods*. Cornell University Baskaran, Balaji, and Anca Ralescu. 2016. A Study of Android Malware Detection Techniques and Machine Learning. Proceedings of the 27th Modern Artificial Intelligence and Cognitive Science Conference, 15-23.

[71] Biau, G´erard. 2013. *Analysis of a Random Forests Model.* Journal of Machine Learning Research, 1063-1095. Bishop, Christopher. 2006. Pattern Recognition and Machine Learning. New York: Springer Bremer, Jurriaan. 2015.