uc3m | Universidad Carlos III de Madrid

University Degree in Computer Science and Engineering
2019-2020

Bachelor Thesis

"Software for malicious macro detection"

Miguel Pedro Peidro Paredes

José María De Fuentes García-Romero de Tejada

Colmenarejo, July 2020

RESUMEN

El objetivo del presente trabajo es ofrecer un estudio detallado del proceso de desarrollo de una herramienta para la detección del virus Emotet en archivos de Microsoft Office, por ser un virus que ha venido causando estragos sobre todo en el panorama empresarial, desde sus inicios como Troyano bancario hasta la actualidad.

Efectivamente esta familia polimórfica ha conseguido generar inconvenientes, evidentes, incalculables y globales en la actividad empresarial sin discriminar por tipología societaria, afectando a cualquier empresa con independencia de su tamaño o sector, inclusive adentrándose en dependencias Gubernamentales, así como, en la propia ciudadanía en su conjunto.

La existencia de dos principales obstáculos para la detección de este virus, constituyen una realidad intrínseca al mismo, de un lado, la ofuscación en sus macros y de otro, su polimorfismo, son piezas angulares del análisis, centrándose nuestra herramienta en hacer frente precisamente a sendos obstáculos, descendiendo al análisis de las características de las macros y a la creación de una red de neuronas que utilice el aprendizaje automático para reconocer los patrones de detección y deliberar su naturaleza maliciosa.

Con el análisis en profundidad de naturaleza de Emotet, nuestro objetivo es formular una serie de características extraíbles de sus macros maliciosas y que permitan construir un modelo de aprendizaje automático para su detección.

Tras el estudio de viabilidad de este proyecto, su diseño e implementación, los resultados que emergen avalan la intención de detectar Emotet partiendo únicamente del análisis estático y con la aplicación de técnicas de machine learning.

Los ratios de detección que muestran las pruebas realizadas al modelo final, presentan una precisión del 84% y únicamente un 3% de falsos positivos durante este proceso de detección.

Key words: Emotet, malware detection, machine learning, botnet, banking trojan.

ABSTRACT

The objective of this work is to give a detailed study of the development process of a software tool for the detection of the Emotet virus in Microsoft Office files, Emotet is a virus that has been wreaking havoc mainly in the business environment, from its beginnings as a banking Trojan to nowadays.

In fact, this polymorphic family has managed to generate evident, incalculable and global inconveniences in the business activity without discriminating by corporate typology, affecting any company regardless of its size or sector, even entering into government agencies, as well as the citizens themselves as a whole.

The existence of two main obstacles for the detection of this virus, constitute an intrinsic reality to it, on the one hand, the obfuscation in its macros and on the other, its polymorphism, are essential pieces of the analysis, focusing our tool in facing precisely two obstacles, descending to the analysis of the macros features and the creation of a neuron network that uses machine learning to recognize the detection patterns and deliberate its malicious nature.

With Emotet's in-depth nature analysis, our goal is to draw out a set of features from the malicious macros and build a machine learning model for their detection.

After the feasibility study of this project, its design and implementation, the results that emerge endorse the intention to detect Emotet starting only from the static analysis and with the application of machine learning techniques.

The detection ratios shown by the tests performed on the final model, present a accuracy of 84% and only 3% of false positives during this detection process.

Key words: Emotet, malware detection, machine learning, botnet, banking trojan.

AKNOWLEDGEMENTS

"El hombre es un ser social por naturaleza" es una frase del filósofo Aristóteles (384-322, a. de C.) para constatar que nacemos con la característica social y la vamos desarrollando a lo largo de nuestra vida.

Quería introducir mis agradecimientos en relación con este trabajo con esta cita considerando que sin las personas que me han acompañado en este viaje, este trabajo no tendría la misma condición. Todas estas personas constituyen una parte fundamental que considero insustituible.

Mi más profundo agradecimiento a José María de Fuente, al que considero mi mentor, por su increíble labor como director del presente trabajo, por mantener un alto grado de exigencia en todas las materias y por su inquebrantable compromiso por explotar al máximo mi potencial. También agradezco la experiencia de haber tenido la oportunidad de aprender bajo su dirección, de mantener discusiones profundas siempre con la mejor de sus intenciones.

Dos personas a las que tengo que agradecer no solo tener la capacidad de realizar este trabajo por los medios, conocimientos y educación que me han brindado, sino por modelar mi persona con cariño, con disciplina y con su mirada atenta, mis padres. Puedo vislumbrar la ética irrompible de mi madre en la intención de este trabajo, la cual agradezco sin medida, por haber luchado por hacer de mí una persona proactiva y por infundir valores de cooperación. Puedo ver también el interés por la técnica, y la disciplina que tanto admiro de mi padre, la conducta de constante mejora hasta alcanzar los objetivos que caracterizan su persona.

Un lugar especial está reservado para la persona que más me ha acompañado en este trabajo, que ha sufrido las inclemencias y el esfuerzo que ha supuesto como nadie motivándome a seguir. A esta persona también va dedicada la cita porque cada día me demuestra que el tiempo está para pasarlo con las personas a las que amas, aunque transcurra delante de un ordenador entre discusiones de trabajo. En este punto y a vista pasada, no sería capaz de volver a hacerlo sin ti y se que el resultado no sería en absoluto el mismo por tu ayuda y por hacerme querer dar lo mejor de mi.

Quiero dar las gracias a las personas que me han acompañado en este viaje, durante el periodo de tiempo que comprende este trabajo han tenido lugar acontecimientos

INDEX

FIGURE INDEX

TABLE INDEX

ACRONYMS AND DEFINITIONS

ACRONYMS

- **API:** Application Programming Interface.
- **C2:** Command and control (server).
- **FT:** False Positive Rate.
- **GUI:** Graphic User Interface
- **IOC:** Indicator Of Confluence.
- **MLP:** Multi-Layer Perceptron (model).
- **PE:** Portable Executable (files).
- **SVM:** Support Vector Machine (model).
- **VBA:** Visual Basic for Applications.

DEFINITIONS

- **Botnet:** group of infected computers controlled by an attacker remotely.
- **Heuristic-based detection:** method of detecting viruses by examining code for suspicious properties. Also known as anomaly-based detection.
- **Hook:** means of executing custom code (function) either before, after, or instead of existing code.
- **Machine learning:** application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
- **Macro:** A series of instructions that are stored so that they can be executed sequentially by a single call or execution order.
- **Malspam:** malicious spam. Popular and effective method for delivering emails in bulk that contain infected documents or links that redirect users to websites that contain Exploits Kits.
- **Malware:** Software that performs harmful actions on a computer system intentionally and without the user's knowledge.
- **Man-in-the-Browser attacks:** Form of Internet threat related to man-in-the-middle (MITM). It is a proxy Trojan horse that infects a web browser by taking advantage of vulnerabilities in browser security to modify web pages, modify transaction content or insert additional transactions, all in a completely covert fashion invisible to both the user and host web application.
- **Payload:** Portion of the malware which performs malicious action**.**
- **Phising:** Fraudulent attempt to obtain sensitive information by disguising oneself as a trustworthy entity.
- **Polymorphism:** Ability of an object to take on many forms.

- **Powershell commands:** Powershell is a Windows operating system tool that provides direct communication with the system by executing operating system commands.
- **Sandbox:** a virtual space in which new or untested software or coding can be run securely.
- **Signature-based detection**: anti-malware approach that identifies the presence of a malware infection or instance by matching the code pattern of the software in question with the database of signatures of known malicious programs, also known as blacklists.
- **Social engineering**: In the context of information security, psychological manipulation of people into performing actions or divulging confidential information.
- **Spam:** Unsolicited e-mail sent to a large number of recipients for advertising or commercial purposes.
- **Trojan:** Apparently legitimate and harmless program, but when executed, gives an attacker remote access to the infected computer.

1. INTRODUCTION

1.1. MOTIVATION

According to data published on the Kaspersky report, 975,491,360 attacks were detected online in 2019. (Kaspersky, 2019)

"*Median company received over 90% of their detected malware by email […] 76% of these attacks are motivated by financial gains and the average cost of an attack is estimated at $ 1 million."* (Verizon, 2019)

For all these reasons, attacks that spread through email constitute a considerable attack vector that affects both the economic level, the infrastructure of all types of companies and even the trust of these companies. Any effort to prevent these attacks, therefore, will directly benefit these factors.

"Emotet, a variant of the Feodo trojan family, first emerged in 2014 as a threat designed to steal banking credentials and other sensitive information. It is most often propagated by phishing emails containing an infected document or malicious website link. […] If there is one threat that dominated 2018 in terms of propagation and persistence, it is Emotet." (Cylance, 2019)

This virus presents a high degree of polymorphism, this means that it changes its structure notably which allows it to evade traditional antivirus products and signature-based detection.

The aim of the project is to check if the Emotet virus can be detected only from static analysis with the application of machine learning techniques to build an agile detection method without running the samples.

In this way, we want the result of the work to answer the following question: is it possible to detect Emotet using static analysis and machine learning models for classification?

Lately, with the increasing use of email, it became an element that caught the attention of cybercriminals, who began to exploit it with malicious purposes to take advantage of users. Since then, it was possible to identify a myriad of campaigns that used email messages and attachments to compromise potential victims.

There is no doubt that these threats are growing at an exorbitant rate, and taking into account that email services are fundamental elements in our daily lives, developing a tool

that provides greater security to users of these services and to the infrastructure, is certainly a very valuable task.

In addition to the personal satisfaction for the contribution to the community, it is a great opportunity to apply the knowledge acquired during this academic period and obtain recognition from them.

1.2. PROJECT OBJECTIVES

One of the main obstacles when it comes to detecting malware is the multiple ways in which it is presented. We know that not only are there different families of malware, but also, that the same family can take on different forms and constitute a significant challenge for detection tools.

The aim of this work is to develop a detection tool capable of identifying the appearance of one of the types of malware that is currently causing the most impact. The first step in Emotet detection process is to gain as much knowledge as possible about this type of threat, its infection and propagation techniques and the damage that can be avoided by detecting it.

This tool will find the presence of Emotet evidence in Microsoft Office files attached to emails before they are opened and executed in order to prevent the infection of systems and the spread of this threat.

1.3. STRUCTURE OF THE DOCUMENT

The document is composed of several sections, each of them addressing different aspects of the work.

The state of the art is a preliminary analysis of the relevant factors for the project such as the context, **the regulatory framework**, an introduction of the Emotet virus and **the socio-economic impact** that this tool could provide.

An in-depth analysis of Emotet. In this section, we detail Emotet's technical features, the actions it performs during each phase of the attack as well as the methods that currently exist to detect it.

Afterwards, a section about the contribution made. This chapter details the theoretical framework of the idea to be developed: the process of collecting documents for the study, the characteristics that will be used during the analysis and the machine learning models for classification.

Project management explains the processes that have been followed to ensure the successful development of the project detailing aspects such as requirements specification, planning, design, analysis, **budget** ...

The implementation chapter details the development of the proposed solution and the tests performed to verify functionality.

In the results analysis section the capacities of the final's model tool are evaluated.

Finally, the conclusions and future work lines are exposed.

2. STATE OF THE ART

This chapter is divided into several parts: first, a brief explanation of the context in which our project is developed, the regulatory framework, the socio-economic impact and the existing precedents related to this work.

2.1. CONTEXT

Due to the constant demand for cyber-threat detection methods, in particular the detection of Emotet virus attacks, not only companies are proposing solutions to the problem.

"The Japan Cyber Emergency Response Team (JPCERT) has launched a new utility, aimed for Windows operating system users, which allows them to easily check whether or not a computer is infected by Emotet Trojan type malware". (JPCERT, 2019)

Internally, companies are making great efforts to warn their workers about possible threats, given that effective attacks are mostly a combination of unawareness and the effectiveness of social engineering techniques used by this virus.

Focusing on the application of Machine Learning techniques for Emotet detection, we must point to the solutions implemented by the company Microsoft in the Windows Defender service for its operating system. This service implements a strategy of detection by layers that presents the following structure:

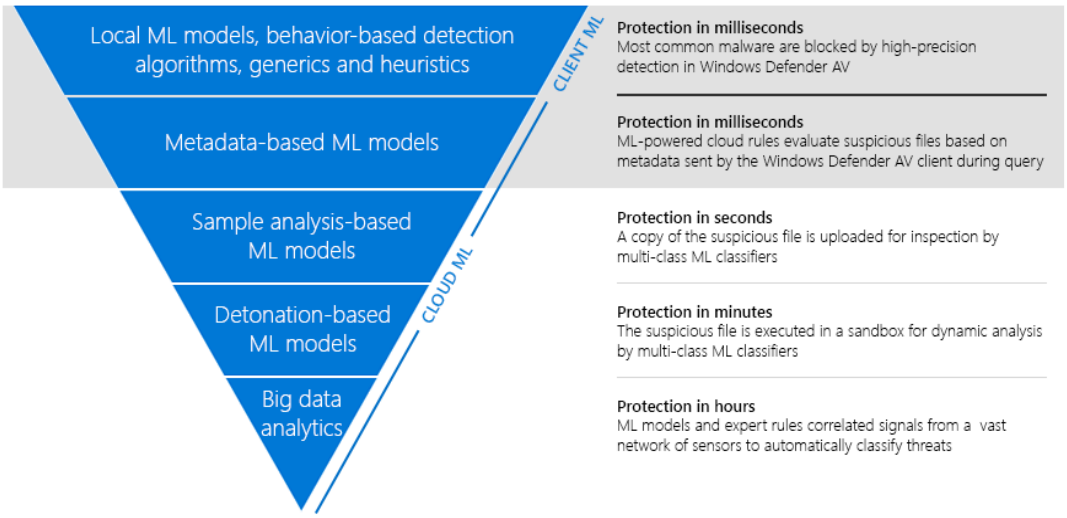


Figure 1. Layered detected model in Windows Defender (Microsoft Defender ATP Research Team, 2018)

These have proven to be effective in detecting this virus: *"In the case of the Emotet outbreak on February 3, Windows Defender AV caught the attack using one of the PE*

gradients boosted tree ensemble models." (Microsoft Defender ATP Research Team, 2018).

On the other hand, the IT security services provider company ESET implements a detection model based also on classification models fed by static analysis, dynamic analysis and memory analysis in sandboxing environment, which presents the following structure:

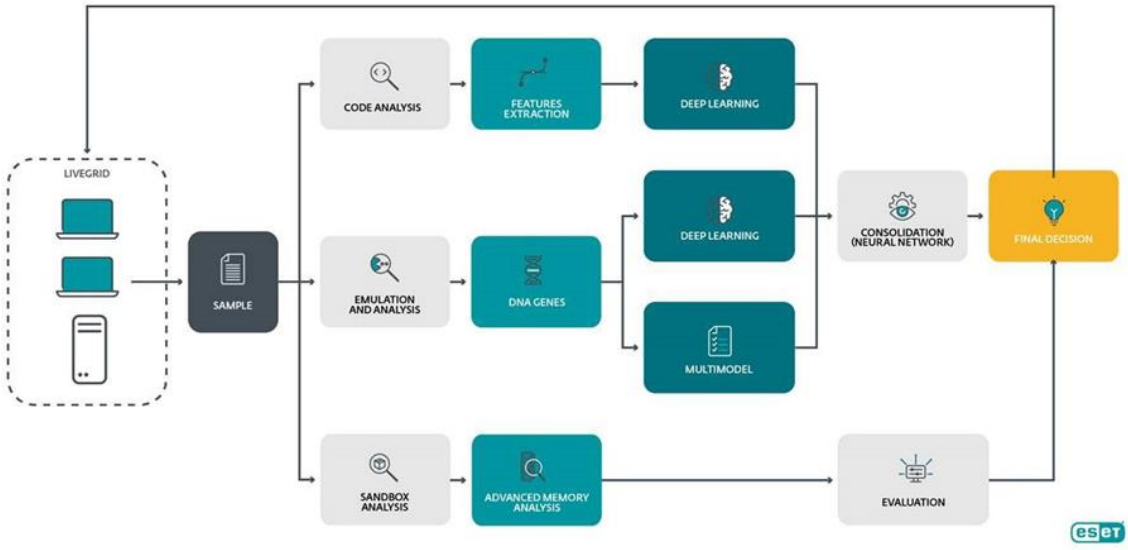


Figure 2 ESET detection model (ESET, 2019)

2.2. REGULATORY FRAMEWORK

Cybersecurity can therefore be broken not only by the commission or omission of certain acts that have to do with security in itself, but sometimes the right of a third party can also be affected by taking advantage of acts that go specifically against the security of a network

In a global world, recent waves of cyber-attacks affect everyone, both at the business level and at institutions of special relevance. It is in this scenario where legal intervention is necessary for the regulation of crimes that are committed through computer systems, the Internet and other information and communication technologies. The so-called "cyber crimes", the new 2.0 crimes.

In order to deal with the condemnation of cyber-crime, which is where we must frame the possible damage to be done by the "Emotet" virus, which can generate a criminal phenomenon and therefore, with consequences for the commission of a criminal act with the peculiarity that the Internet is the medium for such commission, the States typify

cyber-crime in their Criminal Codes, as is the case of Spain, or create special criminal laws, as in the case of the USA or the United Kingdom.

However, cyber-crimes present their own problems, such as the difficulty of determining the place where the illegal acts were committed, which is a necessary issue for the jurisdiction to be applied.

In addition, we may encounter problems in locating cyber-criminals and in obtaining evidence of these crimes, deficiencies in the classification of certain types of behavior and, in short, the effect that police investigation on the Internet has on the fundamental rights of citizens.

As you can see, condemning the infection through our e-mail of a virus of this nature is not easy. Especially when we add that the Internet has a decentralized and universal character, with a great possibility of anonymity, with an open nature, the inexistence of a sovereign or authorities with sufficient powers to discipline all the activity that takes place in cyberspace.

Globalization not only entails short-sighted regulations based on a country's geographical barriers, but it also seeks universal postulates that would unify penalties for objectively illegal acts.

This explains the existence of international instruments such as the Convention on Cybercrime, adopted in Budapest on 23 November 2001, and the EU, for its part, has adopted various directives to harmonize regulation within the EU.

In the European Union, Directive (EU) 2016/1148 of the European Parliament and Council, of July 6, 2016, on measures to ensure a high common level of security of networks and information systems in the Union, known as the Directive on cyber security, was transposed into Spanish law by Royal Decree-Law 12/2018, of September 7, on the security of networks and information systems.

After describing the European regulations that affect us as a Member State, in Spain we also have specific regulations that through the Code of Law on Cybersecurity, published in the Official State Bulletin of August 5, 2020, which contains the most important rules to be taken into account in relation to the protection of cyberspace and to ensure cyber security. .[1]

[1] Several documents and web pages have been used to develop this part of the document. All the sources are noted in the bibliography section.

Also worth mentioning is the Criminal Code (Organic Law 1/2015, March 30), specifically Articles 197 and 264, respectively.

On one hand, Article 197, which condemns illegal acts related to the discovery and disclosure of secrets: it envisages, the seizure of letters, papers, email messages or any other documents of a personal nature from the victim and also, the interception of any type of communication from the victim, whatever the nature and route of such intercepted communication, without the victim's consent.

According to the preamble of LO 1/2015, of 30 March, a response is also offered to cases in which images or recordings of another person are obtained within the scope of their privacy and disclosed against their will, injuring such privacy.

What is intended to be protected by the classification of these actions is privacy, a fundamental right of Article 18 of the Constitution. This right comprises two different dimensions; on the one hand, bodily intimacy, and on the other, personal intimacy.

On the other hand, Article 264 punishes the conduct of data, computer programs, or electronic documents of others, while those related to the normal operation of a computer system of others are punished in the new Article 264 bis.

In addition, Article 264.2 aggravates the penalties in the case of criminal acts committed within the framework of a criminal organization, when particularly serious damage has been caused, or when the general interests have been affected.

Finally, it should be noted that cybersecurity includes not only the commission or omission of certain acts that have to do with security in itself, but sometimes the right of a third party can also be affected by the use of acts that specifically go against the security of a network, so that even more specific aspects are regulated and all of them must be complied with, paying special attention to the Intellectual Property Laws and the Data Protection regulations, when the violation of a work that can be protected by law or the violation of data, respectively, has occurred collaterally.

That said, we note that the presence of Emotet in the user's email attachments, could suggest the possible application of a criminal offence in which both the Criminal Code, as the Law on Intellectual Property (Law 2/2019 of March 1), as the Data Protection Act (Law 3/2018 of December 5), depending on the extent and seriousness of the facts.

We understand, once the regulatory framework has been analyzed, that the prevention of this type of behavior must have a clear technical dimension, not only legal, so that a challenge for the technology itself would be to progressively offer solutions to the risks

it generates: antivirus software and security and intrusion detection systems would undoubtedly be part of this type of measure.

Therefore, the development of our tool can contribute with its implementation to be one of the technical measures of security and therefore, to become with others, a timely way to prevent illegal behavior, having even a much more effective than the fear of a criminal penalty.

2.3. EMOTET

Emotet is a complex modular and polymorphic banking Trojan detected in 2014 that launches large spam campaigns using botnets, attaching malicious links or different types of files such as PDFs, executables or Office documents that inject malicious code in the form of VBA macros and using system calls, execute system commands or powershells to install unwanted programs.

As the techniques used by this type of malware are very broad and diverse, we will devote a section to explaining the different infection and dispersal processes carried out by Emotet.

The interests of these attacks are mainly the theft of bank credentials or access to company infrastructures, data encryption of infected computers to obtain rewards for the recovery of these, inclusion of the computers into your botnet and propagation via email or even in the most advanced versions through local networks or WIFI.

“Emotet continues to be among the most costly and destructive malware affecting state, local, tribal, and territorial SLTT governments, and the private and public sectors. And the infections have cost SLTT governments up to $1 million per incident to remediate.” (Cybersecurity and Infraesctructure Security Agency, 2018), this is how US Department of Homeland Security introduces to us Emotet.

But governments are not the only targets of Emotet. The following graph shows the proportion of the sectors affected by the Emotet virus according to the "2019 Threat Report" of the CYLANCE Company.

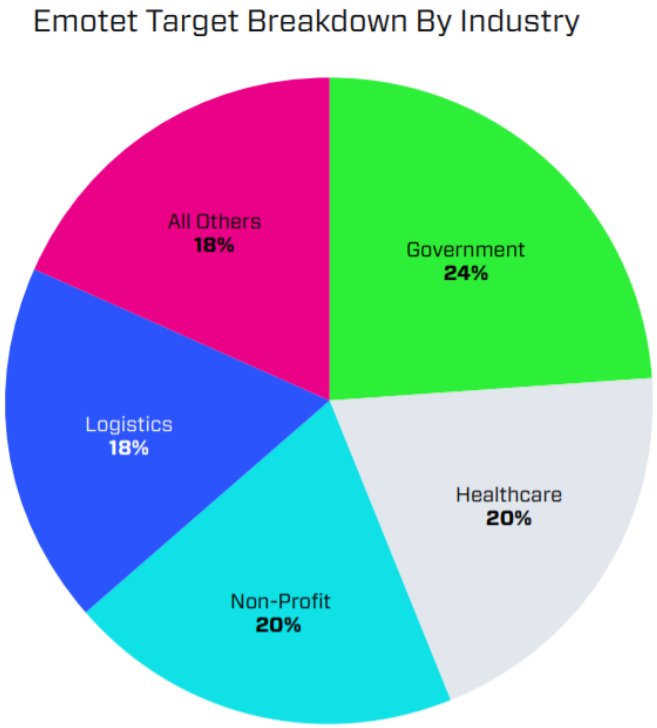


Figure 3 Emotet Target Breakdown by Industry. (Cylance, 2019)

2.3.1. CAPABILITIES

Originally Emotet was a banking Trojan designed to steal financial information from online banking sessions through man-in-the-browser (MITB) attacks, but since 2017 it has been observed distributing other malware families, such as IcedID, Zeus Panda, TrickBot and many other.

In 2020, Emotet is one of the main threats detected around the world. This finding is supported by data from the Center for Internet Security (CIS) indicating that Emotet is one of the most prevalent malware families currently being distributed.

Emotet has the following capabilities:

- Download and run other families of malware, typically banking Trojans
- Brute force attacks on weak passwords using a built-in dictionary
- Steal credentials from web browsers and email clients using legitimate third-party software, specifically NirSoft Mail PassView and WebBrowserPassView
- Steal network passwords stored on a system for the current logged-on user using legitimate third-party software, namely NirSoft Network Password Recovery
- Steal email address books, message header and body content
- Send phishing campaigns from hosts that are already infected, i.e. the Emotet botnet
- Spread laterally across a network by copying and executing itself via network shares over Server Message Block (SMB) protocol

Emotet has several anti-analysis features, designed to frustrate detection of the malware:

- A polymorphic packer, resulting in packed samples that vary in size and structure
- Encrypted imports and function names that are deobfuscated and resolved dynamically at runtime
- A multi-stage initialization procedure, where the Emotet binary is injected into itself
- An encrypted command and control (C2) channel over HTTP. Version 4 of Emotet uses an AES symmetric key that is encrypted using a hard-coded RSA public key. Older versions of Emotet encrypted the C2 channel using the simpler RC4 symmetric-key algorithm

This list has been taken from the Technical Analysis of the Bromium Company report. (Bromium, 2019)

2.4. SOCIO-ECONOMIC IMPACT

After the study of Emotet's capabilities and power to destroy and disrupt users lives. It is needed to study the socio-economic impact an Emotet detector can cause. It is known that everyone can be a target for Emotet. As of today, Emotet has hit companies, individuals, and government entities across the world, with a big incidence in Europe and the United States, stealing from banking logins to financial data.

In January 2020, Emotet occupied the first place in the list of the most prominent data harvesting malware families (INTERPOL, 2020)

Emotet infections have cost SLTT governments up to $1 million per incident to remediate. (Cybersecurity and Infraesctructure Security Agency, 2018)

A July 2019 Emotet strike on Lake City, Florida cost the town $460,000 in ransomware payouts. (Mazzei, 2019).

"One noteworthy Emotet attack on the City of Allentown, PA, required direct help from Microsoft's incident response team to clean up and reportedly cost the city upwards of $1M to fix." (Sheehan, y otros, 2018)

All these facts set the precedent to develop a tool attempting to avoid something similar to happen again.

Now that Emotet is being used to download and deliver other banking Trojans, the list of targets is potentially even broader. Early versions of Emotet were only used to attack banking customers in Germany. Later versions of Emotet targeted organizations in Canada, the United Kingdom, and the United States, combining Emotet, Ryuk and TrickBot malware.

There is a clearly positive socio-economic impact of this project. Only with a program which developing may cost around 40.000 euros, as shown on the budget on chapter V, business, government dependencies and even households can save millions of euros. However, the impact is not only monetary. This virus steals login credentials from banks and emails. It gets into the user privacy and can be used to get sensitive information. A tool to evade it is an asset for any computer that must be developed.

3. EMOTET IN-DEPTH ANALYSIS

3.1. TECHNICAL ANALYSIS

The first step to combat Emotet virus is to know in detail all the infection techniques used. This section aims to unravel the strengths and weaknesses of each infection stage, trying to provide a detailed view of a complete Emotet attack.

Emotet attacks can be divided into multiple stages due to the many actions they perform. The aim is to describe each one to give a first overview and guide in the analysis.

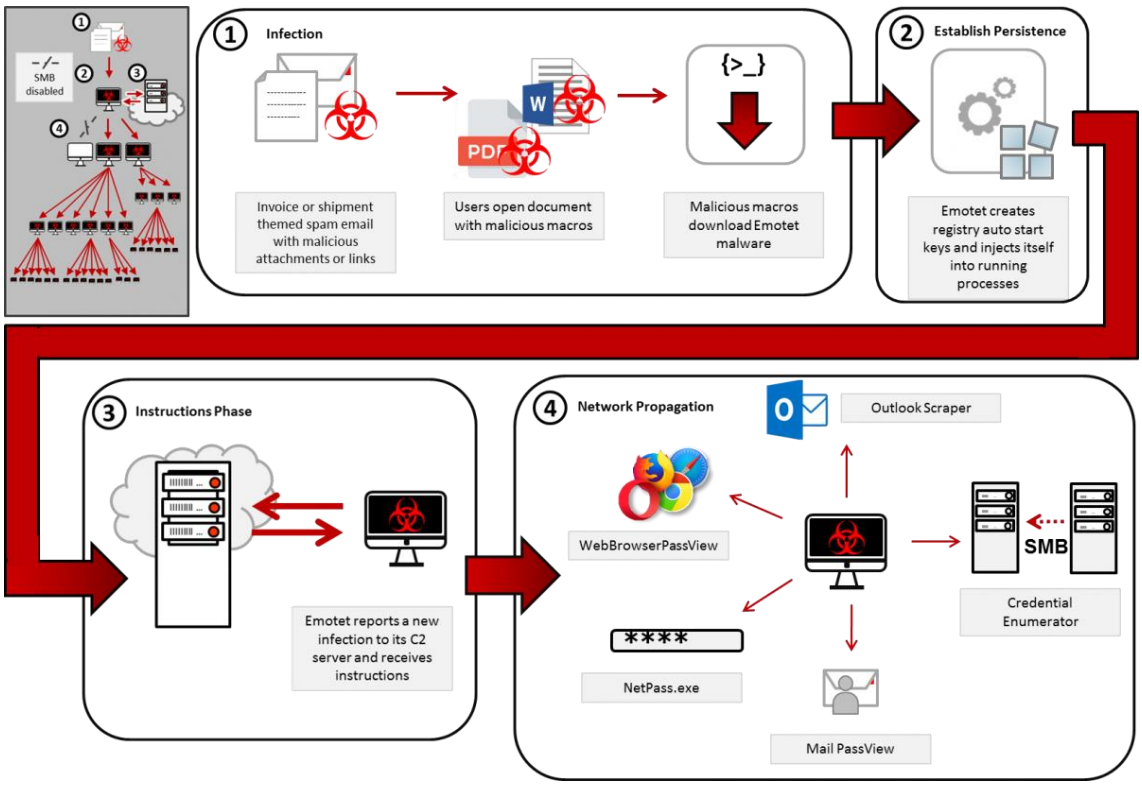


Figure 4 Emotet infection process (Cybersecurity and Infraesctructure Security Agency, 2018)

As shown in this schema, there are four main stages during an Emotet virus attack:

3.1.1. INFECTION

The Emotet attacks usually start with the delivery of an attached document in an email. These messages use multiple social engineering techniques to persuade the user and make an effective attack.

With the opening of the attached documents, it is common to see a customization layer for **macro enablement**. This step is necessary for the virus infection.

Attached documents often have the following formats, with Microsoft Office documents being the most common:

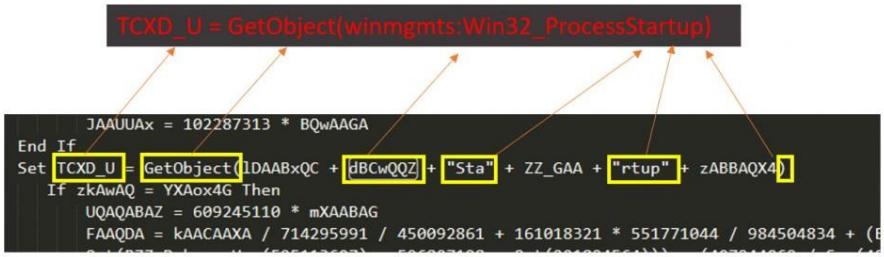
Table 3-1. Most common Emotet infected files formats

FORMAT	NOTES
Microsoft Word 97-2003 Document (.DOC)	Delivered as attachment or hyperlink in a phishing email. Relies on VBA AutoOpen macro for execution. Downloads loader using WebClient.DownloadFile method
Microsoft Word XML Document (.XML)	Delivered as attachment or hyperlink in a phishing email. Relies on VBA AutoOpen macro for execution. Downloads loader using WebClient.DownloadFile method. Renamed with .DOC file extension
Office Open XML Document (.DOCX)	Delivered as attachment or hyperlink in a phishing email. Relies on VBA AutoOpen macro for execution. Downloads loader using WebClient.DownloadFile method. Renamed with .DOC file extension
JavaScript	Delivered in ZIP file attached to a phishing email or hyperlink in PDF. Downloads loader using MSXML2.XMLHTTP object
Portable Document Format (PDF)	Delivered as attachment in a phishing email. Contains hyperlink to Word document or JavaScript downloader

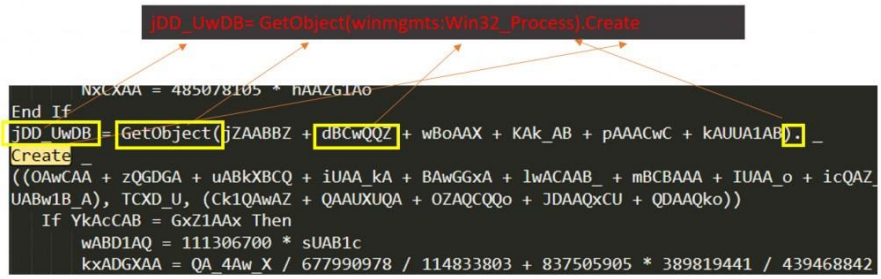
The documents have several embedded VBA macros that execute an AutoOpen function, automatically triggering a series of malicious instructions.

"The VBA code shown below references Windows Management Instrumentation (WMI) classes winmgmts:Win32_ProcessStartup and winmgmts:Win32_Process. On execution, the AutoOpen subroutine uses these WMI classes to launch an instance of PowerShell that runs a Base64 encoded command in the background" (Bromium, 2019)

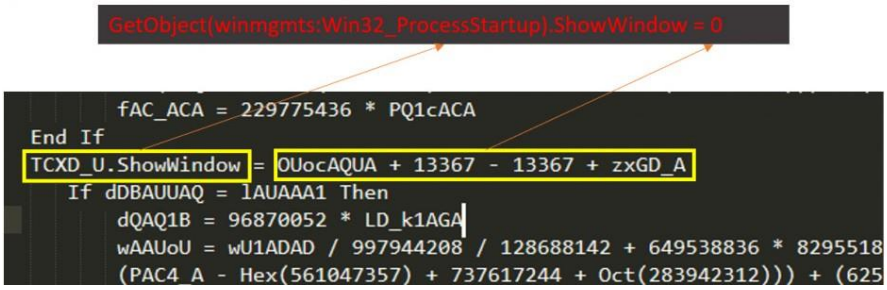
Figure 5. Emotet WMI macros (Bromium, 2019)



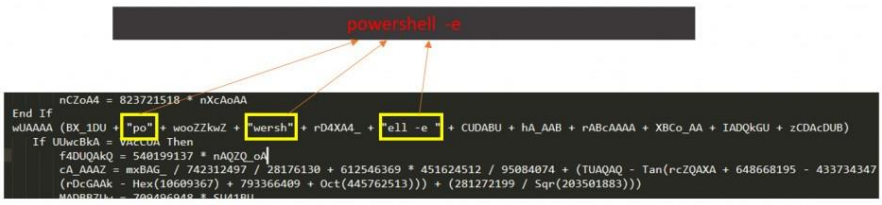
Variable TCXD_U is defined with the string “Geinmgmts:Win32_ProcessStartup)”



Variable jDD_UwDB is defined with the string GetObject(winmgmts:Win32_Process).Create”



Sets the parameter of “GetObject(winmgmts:Win32_ProcessStartup).ShowWindow” to a value of 0



Creation of string “powershell -e”

"The process is launched as a child process of WmiPrvSe.exe (WMI Provider Host). Launching PowerShell this way benefits the malware operators because they are more likely to evade process chain-based detection" (Bromium, 2019)

The powershell commands, in addition to being coded with base 64, have a similar obfuscation to that of the macro code. This technique makes it difficult to identify the instructions and evades virus detection techniques.

The execution of the powershell command performs the Emotet loader download. The downloaded file is saved to the victim's user profile directory (typically C:\Users\[Username]) with two or three digit filename.

3.1.1. ESTABLISH PERSISTENCE

The Emotet Loader file presents a hiding structure provided by customized packers, which makes the static analysis of the binary code difficult. This process embeds the executable inside another executable.

The process starts when the powershell instance creates a child process with the Emotet loader image. This process is the container of the binary code packed in the packing process. Therefore, the Emotet loader creates three processes with a tree structure that can be seen in the following image:

Figure 6. Emotet establish persistence: Tree process (Lu, 2019)

You can see how the execution time of the first process is very reduced since it only creates the child process that includes the packed code.

The second process is copied to the C:\Windows\SysWOW64 address with a different name and generates a service that points to the created executable.

This service will be started by services.exe and will use the same technique as in the previous case. The process that is started creates a child process in turn, which downloads the next phase of the payload by overwriting the first executable.

Below is the perspective of the memory during the initialization process of the Emotet Loader taken from the report of the specialized malware detection service of the Bromium company.

"*In summary, the unpacking and initialization procedure for the Emotet loader follows these steps:*

1. *The dropped Emotet binary (15.exe) allocates a new memory region with execute permission and writes a code stub there (memory region 1).*
2. *The stub decrypts an embedded PE file from the .data section of the image and writes it in the new memory region (memory region 2).*
3. *The file written to memory region 2 is a valid PE file that is another Emotet binary and can be dumped and executed without needing to fix its relocations.*
4. *The stub from memory region 1 allocates a new region with execute permission (memory region 3).*
5. *The stub reads an embedded payload from memory region 2 and writes it to memory region 3.*
6. *After writing the payload to memory region 3, it then modifies it by inserting new code and trampolines.*
7. *Once the payload is ready in memory region 3, it unmaps the 15.exe image.*
8. *After unmapping the image, it allocates a new region of the same size as memory region 3 with execute permission and copies the payload from memory region 3 to the newly allocated region (memory region 4).*
9. *The stub then passes execution to memory region 4, which launches the main Emotet loader*." (Bromium, 2019)

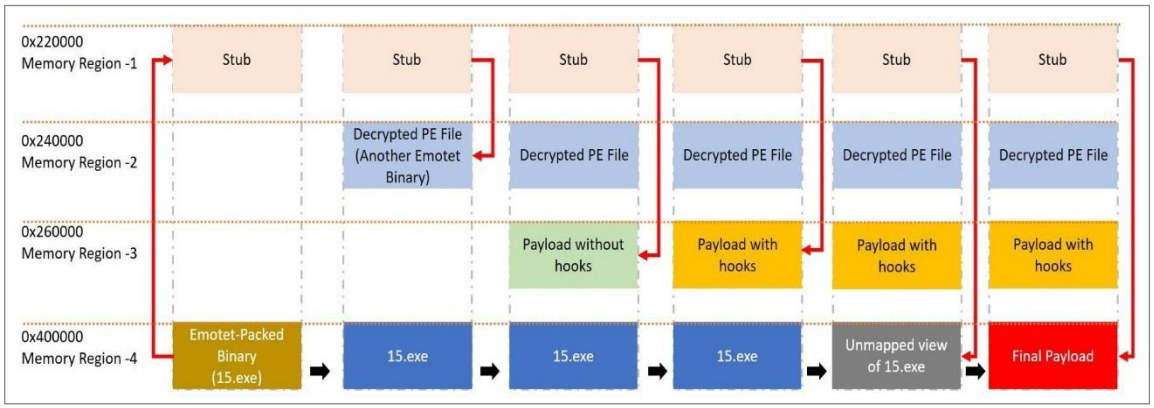


Figure 7 Emotet Loader Memory perspective (Bromium, 2019)

The new Emotet loader collects system information and sends it through an encrypted channel to its command and control (C2) servers.

3.1.2. INSTRUCTION PHASE

Emotet mainly uses the HTTP protocol for communication with command and control servers.

Among the information sent to the C2 server, we find the system specifications needed to establish persistence, the list of processes running on the system, credentials extracted from the infected system, contacts for virus dispersion, etc.

The serialization of this data is done using protobuf. (1)

Data encryption is carried out as follows:

"The Emotet binary contains an RSA public key, which it uses with the CryptGenKey function call to generate an AES 128 symmetric encryption key pair.

Emotet encrypts the data block using this key, then encodes the encrypted data in base64, and finally transmits it by performing an HTTP GET request to the root directory of the C2 server." (Sophos, 2019)

The ports used for this communication are usually 80 for HTTP and 443 for HTTPS. Additionally, a graph is shown with the ports used for communication with command and control servers.

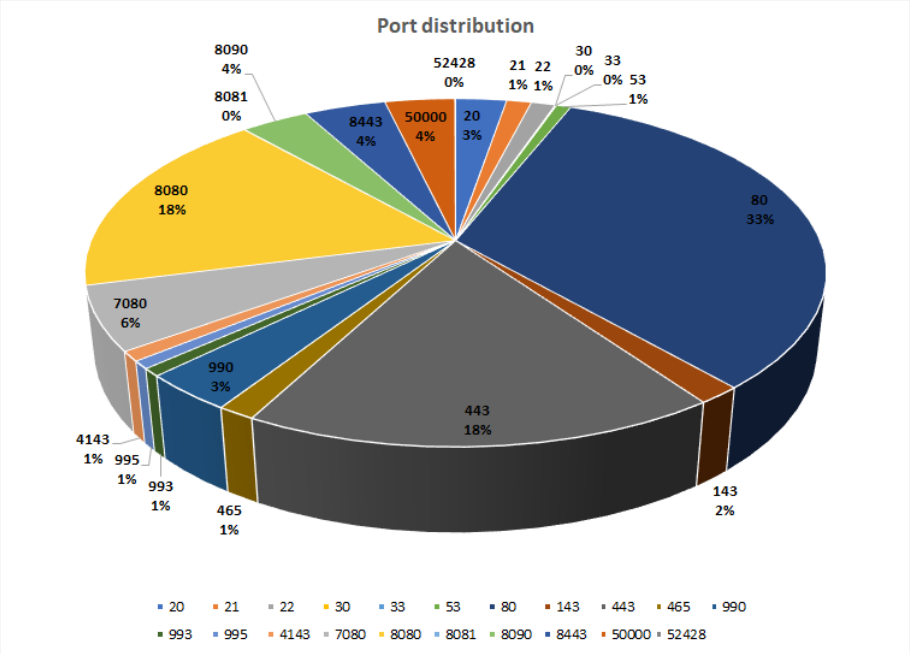


Figure 8. Emotet ports for communication with C2 (Sophos, 2019)

3.1.2.1. NETWORK PROPAGATION

To understand the spread of the Emotet virus, we must know how its infrastructure is built.

In the first versions of the virus, there were servers specialized for the distribution of malspam campaigns. Infected devices would extract email addresses from the accounts hosted on the systems and communicate them to command and control servers to extend the campaign.

From 2019 onwards, it was identified that "*Emotet implements a UPnP module, which allows an infected device to act as C2.”* (Century Link, 2019)

This strategy transforms infected devices into command and control servers, which operate as messengers between the newly infected devices and the higher-level command and control servers that correspond to dedicated web servers.

This builds a decentralized infrastructure and creates an additional routing layer, which prevents identification of Tier 1 C2 and Tier 2 C2 servers.

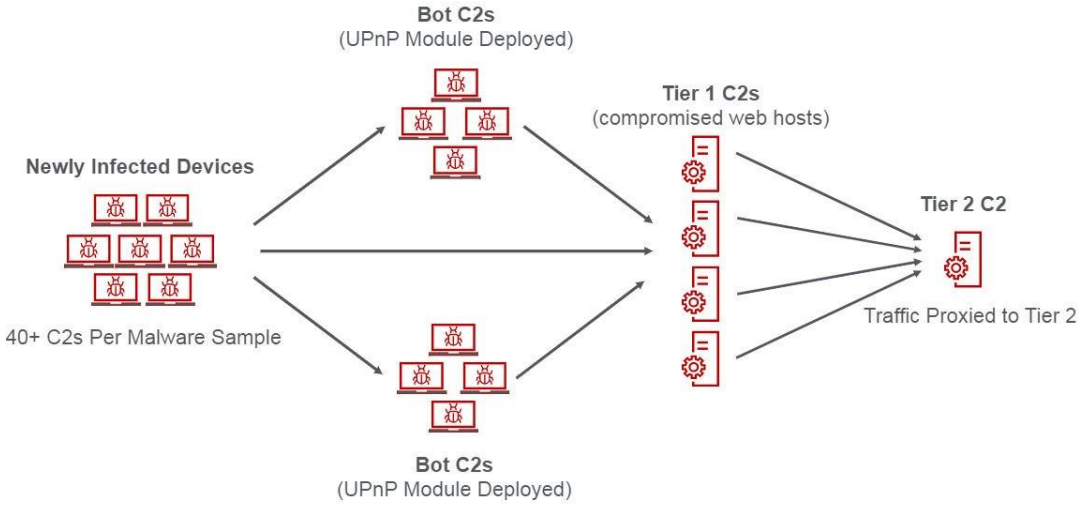


Figure 9. New C2 Servers spreading (Century Link, 2019)

In conclusion, the infected devices send email addresses captured from the infected device to the Tier C2 servers, which include them in the broadcast campaigns, sending malicious emails and completing the propagation chain.

Knowing the techniques used by attackers and implemented by this family of malware, the next step in the design of our Emotet malware detection tool is to gain an in-depth understanding of the techniques used to detect it.

Malware analysis is a process or technique to determine the origin and potential impact of a specific malware sample. There are many approaches to the malware detection problem. Here, we briefly consider signature-based and heuristic-based detection

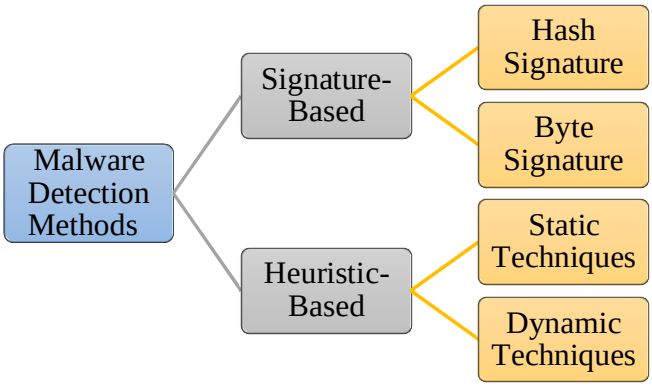


Figure 10 Malware detection methods (Sihwail, y otros, 2018)

3.2. SIGNATURE BASED DETECTION

Signature based detection is the most widely used anti-virus technique. A signature is a sequence of bytes that can be used to identify specific malware. A variety of pattern matching schemes are used to scan for signatures. Signature based anti-virus software must maintain a repository of signatures of known malware and such a repository must be updated frequently as new threats are discovered. Signature based detection is simple, relatively fast, and effective against most common types of malware.

Also, relatively simple obfuscation techniques can be used to evade signature detection, this is the main problem that we find in these malware samples. Due to Emotet polymorphic nature, traditional signature-based methods are not likely to be effective.

This approach extracts unique signatures from captured malware files and use this signature to detect similar malware, this method has small false positive (FP) rate and due to the speed of hash comparison, it could be used in the first instance to perform a quick detection of infected documents, but due to the low efficiency and low detection rate that it presents in detecting these malware families, it would be an ineffective development.

3.3. HEURISTIC BASED DETECTION

Heuristic-based is also known as anomaly or behavior- based detection. This technique finds malicious evidences in source code and the activities performed by malware during runtime, this approach is capable to detect unknown malware and malware that uses obfuscation techniques. For this reason, heuristic based is the better approach to analyze Emotet malware samples.

3.3.1. STATIC ANALYSIS

This technique refers to analyzing the Portable Executable files (PE files) without running them.

In order to show the techniques currently used for Emotet detection, their advantages and disadvantages, and finally to determine our analysis approach, we have compiled all the static and dynamic techniques:

OBJECT TO ANALYSE
VBA Macro Microsoft Office file

Table 3-2. Static analysis

Category	Technique	Description	Advantages	Disadvantages
WMI CALL DETECTION	API Calls scanning	Malicious VBA code in the document will utilize WMI to launch powershell encoded code and then download the second stage payload, the objective is finding this call to evidence Emotet presence	Fast detection No database needed	Deobfuscation needed to find “winmgmts” Polimorphic could evade this method Hard to automate
POWERSHELL COMMAND DETECTION	Power shell command scanning	Previous WMI classes are used to launch an instance of PowerShell that runs a Base64 encoded command in the background. Find instances of PowerShell spawning via WMI searching winword.exe, cmd.exe, powershell.exe	- Fast detection No database needed	Deobfuscation needed to find “winword.exe”, “cmd.exe”, “powershell.exe ”

SYSTEM CALL DETECTION	Creation of Mutexes detection	After executing emotet loader, it creates two mutexes with the format PEM%X. One of the mutexes is created using the parent process ID (PEM[PPID]) and the other uses its own PID (PEM[PID]).	- Fast detection - No database needed	- Deobfuscation needed to find mutex tags
IOC	Compromised **URLs** Comparison	Check URLs extracted from VBA Code with well-known Malicious URL in database	- Fast detection - Easy to extract IOCs using oletools	- Updated database needed - Only well-known samples detected
IOC	Compromised **Domain Name** Comparison	Check Domain Names extracted from VBA Code with well-known Malicious Domain Name in database	- Fast detection - Easy to extract IOCs using oletools	- Updated database needed - Only well-known samples detected
IOC	Compromised **IPs** Comparison	Check IPs extracted from VBA Code with well-known Malicious IPs in database	- Fast detection - Easy to extract IOCs using oletools	- Updated database needed - Only well-known samples detected
IOC	Compromised **MD5 Singnature** Comparison	Check MD5 signature of VBA code to detect well-known Malicious signature in database	- Fast detection - Easy to compare signatures	- Updated database needed - Only well-known samples detected - Polymorphic nature evades easily
IOC	Compromised **Sha265 Singnature** Comparison	Check Sha265 signature of VBA code to detect well-known Malicious signature in database	- Fast detection - Easy to compare signatures	- Updated database needed - Only well-known samples detected - Polymorphic nature evades easily

STRING ANALYSIS	Code Similarity	This method consists in compare VBA source code and detect patterns and similarity with well-known malicious VBA source codes	- Different algorithms to apply - Relatively fast, (data set size) -Deobfuscation not needed	- Updated database needed -High risk to get False positive rate (FP)
STRING ANALYSIS	Average Variable Assignment Length	It was found that many obfuscated macros declared abnormally long string variables. This method determines by comparing the average length for both benign and malicious sets.	- Fast Detection -Deobfuscation not needed - Updated database not needed	- Hard to compute variable length in some cases -High risk to get False positive rate (FP)
STRING ANALYSIS	Count of Integer Variables	Another characteristic common to the malicious macro set was that they defined more integer variables than the benign set. To capture this feature, the count of integer variables in the macro source code divided by the length of the source.	- Fast Detection -Deobfuscation not needed - Updated database not needed	- High risk to get False positive rate (FP)
STRING ANALYSIS	Count of String Variables	The same finding held for string as for integer variables. To capture this feature, the count of string variables in the macro source code divided by the length of the source	- Fast Detection Deobfuscation not needed - Updated database not needed	- High risk to get False positive rate (FP)
STRING ANALYSIS	Macro Keywords	Binary feature that encodes the presence of certain keywords that were found to be significantly more prevalent amongst the malicious set. These keywords related to event-based subroutines described	Fast Detection Deobfuscation not needed	- High risk to get False positive rate (FP) - Updated database needed

		earlier, named AutoOpen, AutoClose, DocumentOpen and DocumentClose		
STRING ANALYSIS	Highest Number of Consecutive Mathematical Operations	High number of consecutive operations is characteristic of malicious scripts	Fast Detection Deobfuscation not needed - Updated database not needed	- High risk to get False positive rate (FP)

3.3.2. DYNAMIC ANALYSIS

With the knowledge of the static analysis techniques we can recognize that it is limited to the analysis of the element obtained in the first instance, in this case it is only focused on the analysis of the source code of the VBA Macro from the Microsoft Office file, but we also know that this is not the only element that we can analyse in an Emotet attack.

The dynamic analysis consists of triggering the attack by running the document in a secure virtual environment (Sandbox) and monitoring its behavior with the system. We have a clear idea of how Emotet behaves in our system, so we can look for evidence in the download of files, communications with C&C servers, etc.

It is also interesting to note that the execution of malware in a secure environment usually results in new scan elements such as executable programs, in which the static scan techniques shown in the previous section can be applied again.

In order to show the dynamic analysis techniques currently applied for detecting Emotet, a compilation of these techniques is presented again in a table:

Table 3-3. Dynamic analisys.

CATEGORY	TECHNIQUE	OBJECT TO ANALYSE	DESCRIPTION	ADVANTAGES	DISADVANTAGES
FUNCTION CALL ANALYSIS	Child process monitor	Process Monitor during VBA Macro execution	A hook implementing a monitoring function is invoked every time an API is called. Kernel and user monitor mode analyze the child processes that can be launched by a malware.	- No database needed	- Virtual Environment needed - High cost of defining hook functions

					- Hard to automate
FUNCTION CALL ANALYSIS	DLL Injection	Operative System Export Address Table (EAT)	Based on code injection of a trusted DLL into the analyzed process, which intercepts function calls by overwriting entries in the Export Address Table. This process collects a vast amount of information such as the name of the function called, its parameters, the system's state, etc.	- Tools like CWSandbox makes this technique easy to apply	- Virtual Environment needed
FUNCTION CALL ANALYSIS	File system monitor	Read/write events on all hard drives	Collects all events triggered by the malware in the File System. Analyze files saved in hard drives	- Low False positive rate (FT)	- Virtual Environment needed - Updated database needed - Results analysis needed
FUNCTION CALL ANALYSIS	Registry monitor	Registry Key	Malware copy itself or its variants on various locations on the file system and then adds a registry key to start automatically with booting, this technique registry operations about OpenKey, CreateKey, DeleteKey, SeteValueKey, etc.	- Low False positive rate (FT)	- Virtual Environment needed - Updated database needed - Analysis of results needed
INFORMATION FLOW TRACKING (IFT)	Data and address tainting	Process Monitor during VBA Macro execution	Taints information like network packets payload, function parameters and data accessed and shows the taint information spread. The analysis is performed at the kernel level to intercept the system calls and APIs are invoked	- No database needed - Tools like Panorama makes this technique so easy to apply	- Several simultaneous programs execution decreases effectiveness -Virtual Environment needed

INFORMATION FLOW TRACKING	TCP Stream	Network Traffic during VBA Macro execution	This technique compare TCP Stream extracted during the VBA Macro execution with TCP Streams of well-known attacks	- Easy to extract TCP Stream (Wireshark)	- Virtual Environment needed - Updated database needed
TRACING	IOCs Extraction	Network Traffic during VBA Macro execution	This technique extract IOCs from network traffic and apply static analysis to find matches	- Fast Extraction - Same database on static analysis	- Virtual Environment needed
TRACING	Volatile memory analysis	Memory state	Provides a perspective of the memory state at a point in the run. Malicious code in kernel instructions can be found during execution.	- Really save way to analyze malware	- Usually no automate detection - Virtual Environment needed - Malware can detect analysis

3.4. CONCLUSION

After collecting the different techniques applied in the detection of Emotet, it is necessary to distinguish the different approaches and capabilities that present the static analysis techniques versus dynamic analysis techniques, we can say that the static analysis has less depth and less scope with respect to the objects of analysis, while dynamic solutions usually present a higher cost of computing and infrastructure. It is also important to note that many of the static analysis techniques have the same detection efficiency and reduce the computational cost and risk of infection of the system. We believe that the best approach for the detection of Emotet is a model that combines the largest number of techniques.

4. CONTRIBUTION

Most of the techniques described on the previous chapter are applied directly by the analysts in the detection process. We have seen that automatic detection tools mainly use methods based on IOCs or signatures and that the main shortcoming in the process of detecting the Emotet virus is the difficulty of applying all the techniques described above in the same detection process.

In order to take a step in this direction, our proposal is based on applying an Artificial Neural Network model: a Multilayer Perceptron capable of discriminating malicious documents containing an Emotet virus sample from the rest based on the results of the largest number of detection techniques.

We will compose a data set of benign and malicious (containing Emotet) Microsoft Office documents to train the Artificial Neural Networks model and test its detection capacity. In this way we will calculate the values for the metrics selected as representative metrics, i.e. presenting differentiated values between benign and malicious samples and we will use them in combination for the training of our model.

We will now explain how this network of artificial neurons works, why is it convenient as a solution to this detection problem and the procedure that we will follow for the model implementation.

4.1. DATA SET

We have seen that one of the main components for the neural network model construction is the set of patterns for the training, validation and test phases. These patterns as defined above are compounded by the results of different metrics on Microsoft Office documents. Therefore, the first step to build our data set is to collect Microsoft Office documents containing VBA Macros, since the analysis of the metrics will be applied to the VBA Macros embedded in the document and belonging to two classes, which will be the classes that will discriminate our neural network: benign and malicious.

4.1.1. BENIGN DOCUMENTS

We call a benign document a document that does not present any evidence of malicious code or activity. The procedure followed to collect these documents consisted on searching for Microsoft Office documents on the Internet using search filters that restrict the document format, thus obtaining a series of Microsoft Office documents related to VBA Macros.

The next step is to check that these documents contain VBA Macros. To do this we have created a folder containing these documents and a script in Python to show the structure

of the documents and evaluate the presence of macros using the open source tool "oledump".

This process has been performed in a virtual environment with operating system "Kali Linux 2.6/64-bit" virtualized in ORACLE VM, creating a safe environment and minimizing the risk of system infection.

We have attached a demonstration of the script's output and the identification of some VBA Macros container documents:

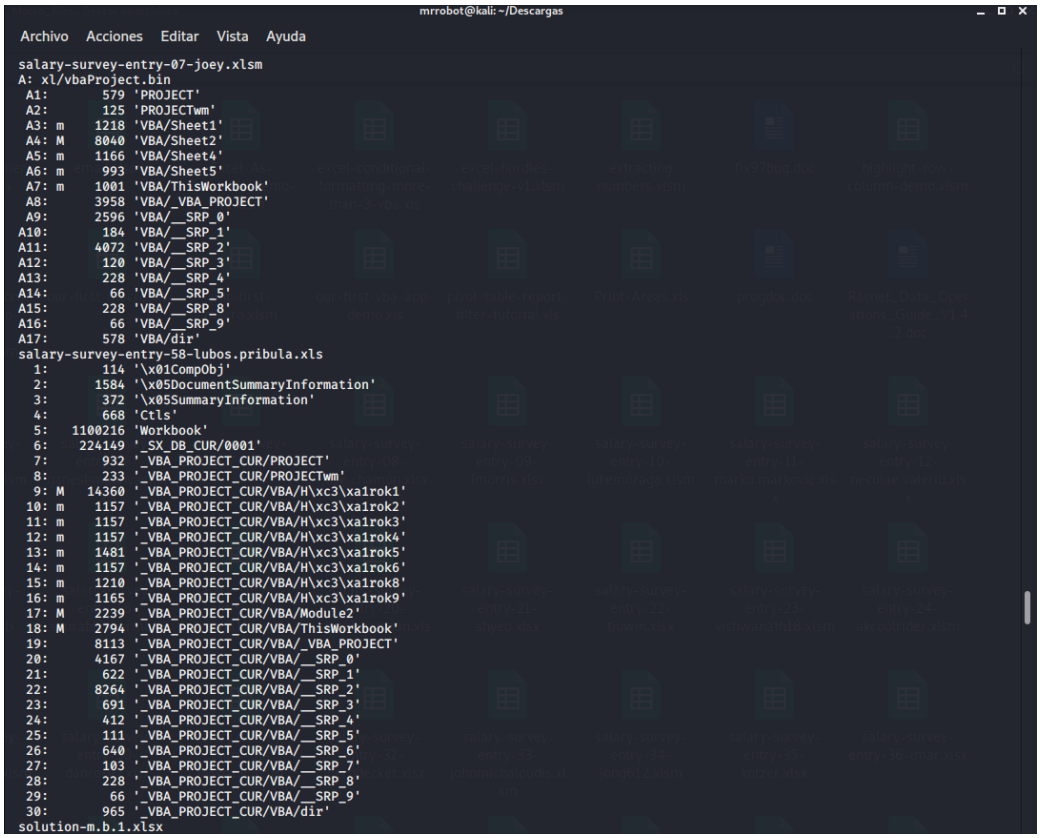


Figure 11. VBA Macros Scheme

In the image you can see the scheme of two documents: "salary-survey-entry-07-joey.xlsm" and "salary-survey-entry-58-lubos.pribula.xls", we can see the VBA Macros modules identified with the letters "M" or "m" and the alphanumeric identifier that precedes it. This way we can discard the documents that do not contain macros since we will not be able to apply any analysis on them.

The next step is to ensure that downloaded documents containing macros are not malicious, as we are trying to create a set of patterns that represent benign documents. To carry out this task we have used the online portal "Virus Total". This portal allows us to analyze documents by applying multiple anti-malware (Cylance, Fortiner ...) and

applying techniques such as: Compromised Domains Comparison (Avira, Kaspersky URL Advisor, ...), Behaviour Analysis (Cuckoo Sandbox, Snort, ...). This way we will use this tool to classify the selected documents as benign if they receive a report with no malware detection.

This image corresponds to the analysis of the document "extracting%20numbers.xlsm" belonging to the data set:

.

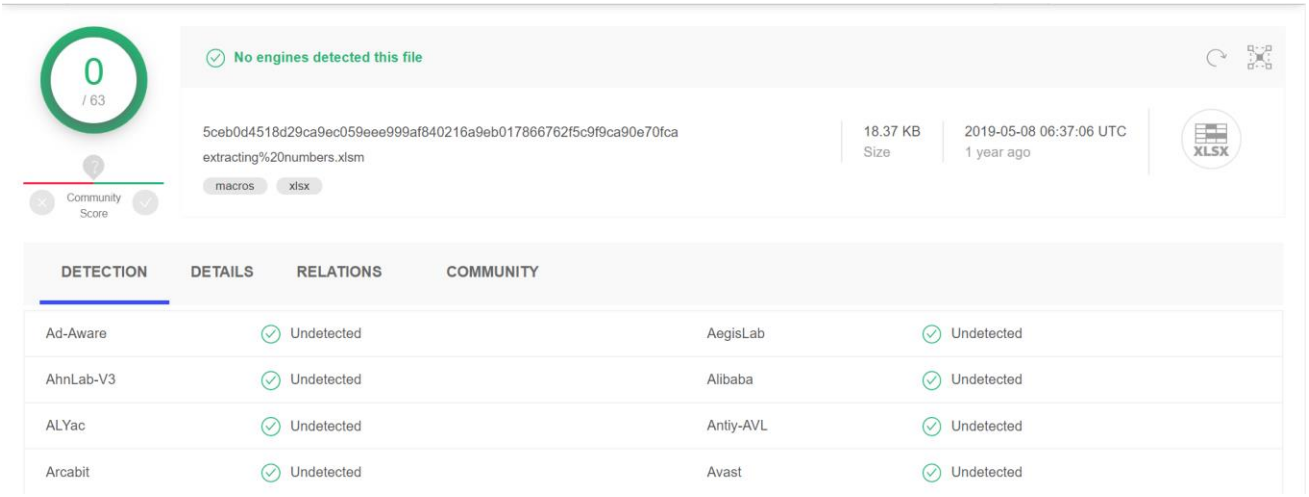


Figure 12. Document analysis

We have performed this analysis process for all files selected as benign in the data set. The decision to assume these documents as benign performing this analysis is due to the size of the data set and the time it would take to perform a more comprehensive analysis. This factor coupled with the fact that later we will perform an analysis of the results of the metrics that we will use as input for the model to identify anomalous patterns or documents that present values scattered to the set, are measures that could involve a more comprehensive analysis but only of specific samples of the data set.

The final result is the selection of 140 Microsoft Office documents, most of which have a .xls or .xlsm extension. There are also .doc and .docx documents, all with VBA Macro container modules and no evidence of malicious code.

4.1.2. MALICIOUS DOCUMENTS

The selection of malicious documents for our data set presents a fundamental factor to be taken into consideration, which is the risk of infection of the system that will drive the selection procedure. Some of the measures we will take have also been implemented in the process of selecting benign documents, given that the risk of downloading documents

from the Internet inherently entails a risk of infection. For this task we have taken the following measures:

- **Virtualized Environment:** Similarly, to the benign document selection, the entire selection procedure has been carried out in a virtualized environment using the ORACLE VM program. We have also implemented security configurations such as locking the USB ports from the virtual machine control panel and restricting the permissions of the virtualization application on our host system to restrict privileged operations by this application.
- **Operating System:** An interesting factor in the process of analyzing Microsoft Office documents is to understand that they are designed to run on a Windows operating system. Using a different operating system gives us an additional layer of isolation since the documents will not be able to run as in their native operating system. For this reason, our virtualized environment is a 64-bit 2.6 Linux distribution, specifically Kali Linux.
- **Connexion with host system:** Since part of the procedure involves downloading documents, it is necessary to establish an Internet connection with the virtual machine, but following analysis recommendations, once the documents have been downloaded, we must disconnect the virtual machine from the host system, thus disabling possible connections to the Internet or directly to the host system. In Virtual Box this is equivalent to remove the network card from the virtualized machine and we can do it in the network settings panel of our machine.

The first step for the selection of Microsoft Office documents containing Emotet samples is to find a source of this kind of documents and collect a significant number of them to start the selection process.

The selected source for this purpose is the community malware search platform "ANY.RUN". This public platform allows users to upload documents and share them in order to determine their nature and allows downloading the malware samples that are analyzed there.

In this way, we carry out a search for the analysis threads concerning the Emotet virus by applying the following filters:

- **Runtype:** “File”, sicne what we want to obtain from the search are Microsoft Office documents, applying this filter we discard URLs, executable programs, etc.
- **Extension:** “Microsoft Office”, following the previous argument.
- **Veredict:** “Malicious”, since we want to obtain the samples in which evidence of the presence of Emotet has been found.
- **Tag:** “Emotet”, specifying that the samples must correspond to the Emotet virus.

At the time of publication, we obtained 401 pages with 50 publications each:

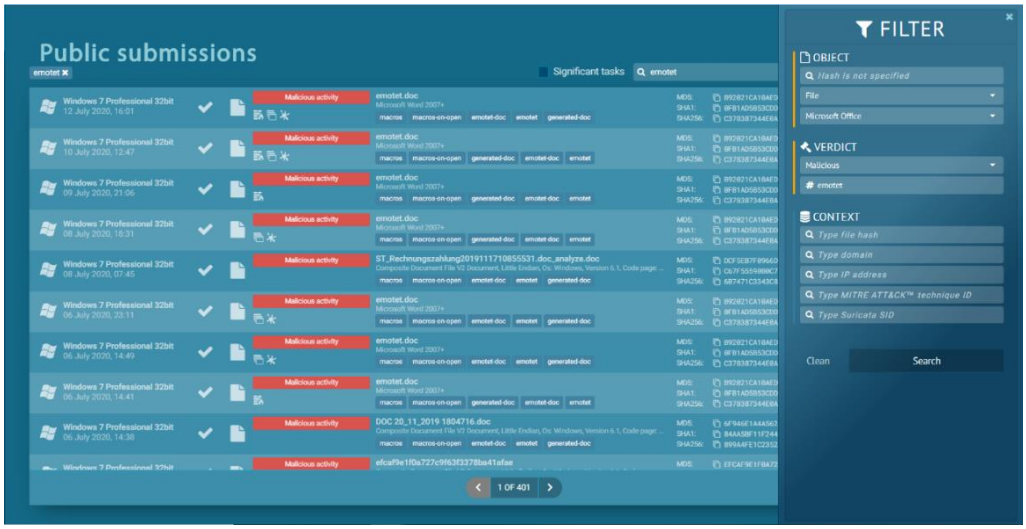


Figure 13 Documents search, list

These documents already record evidence of Emotet presence. Here, we observe analysis of the first result on the search, in which we can observe the record of an WMI Request that executes a powershell command with a coded parameter and this in turn downloads and runs the executable program "ntvdm.exe".

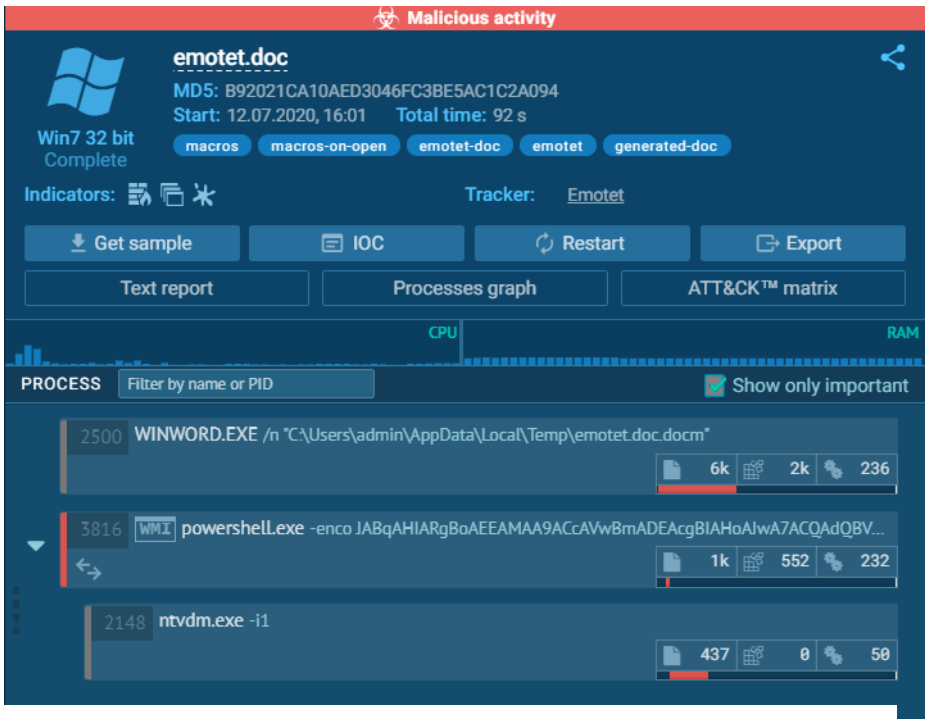


Figure 14. Malicious activity

These evidences help us to know the nature of the samples with which we will later train our model and the reasons why these documents are considered malicious.

For the same reason we do not need to check using other techniques the malicious nature of the downloaded samples as it happened with the selection of benign documents, but as a comparison we will apply the same analysis we used to consider the documents as benign:

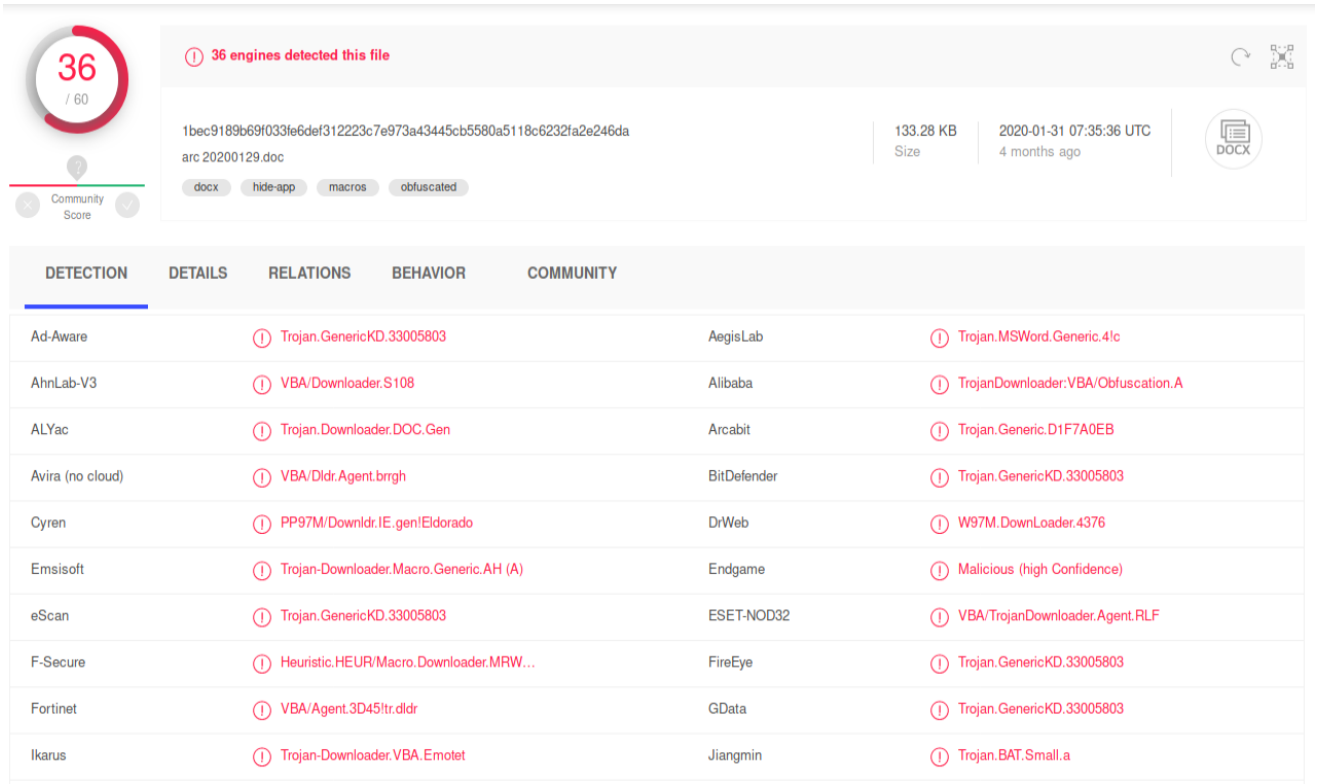


Figure 15. Total Virus analysis

In this case 36 Total Virus engines have discriminated the document "arc 20200129.doc" as being malicious according to the registry of the source platform. This does not provide a guarantee of the benign consideration of the selected set since there are evasive methods that minimize the detection capacity of these tools, but it shows that the tool used is capable of detecting positive samples of the virus.

The result of this process provides 133 malicious documents with their respective evidence for training and testing of the target model.

4.2. FEATURES

Based on the technical analysis of the Emotet virus, we will define the potential characteristics that can contribute to the classification of documents in two classes:

- Malicious: documents suspected of presenting evidence of Emotet's presence.

- Benign; documents that do not present suspicion of malicious behavior.

Due to the concealing intention of the Emotet samples, one of the features identified in figure 3 is the use of concatenation operations with multiple text strings.

We can define the following measures to identify this feature:

- Number of text strings present in the document's macros.
- Average length of identified text strings.
- Number of concatenation operations "+".
- Maximum number of consecutive concatenation operations.

Another evidence that we can extract from the Emotet macros, is the presence of system calls or execution of powershell commands or cmd.

This measure faces the difficulty of identifying text strings in an obfuscated medium as illustrated on the creation of powershell command figure from the previous chapter.

On the other hand, these measures focus on identifying features of malicious documents. By applying a more creative idea, we can select traits that identify benign documents.

Based on the experience gained in selecting documents for the dataset, we have seen that a large number of benign documents containing macros belong to the work environment and contain operations with multiple variables and routines that include loops.

It would not be correct to attribute the presence of variables, or loops, as a benign feature, even though it is a feature absent in the macros of the Emotet samples. However, by way of experimentation they will be selected as discriminating metrics.

4.3. CLASSIFICATION MODELS

4.3.1. MULTILAYER PERCEPTRON

A Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN) that consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

4.3.1.1. MULTILAYER PERCEPTRON ARCHITECTURE

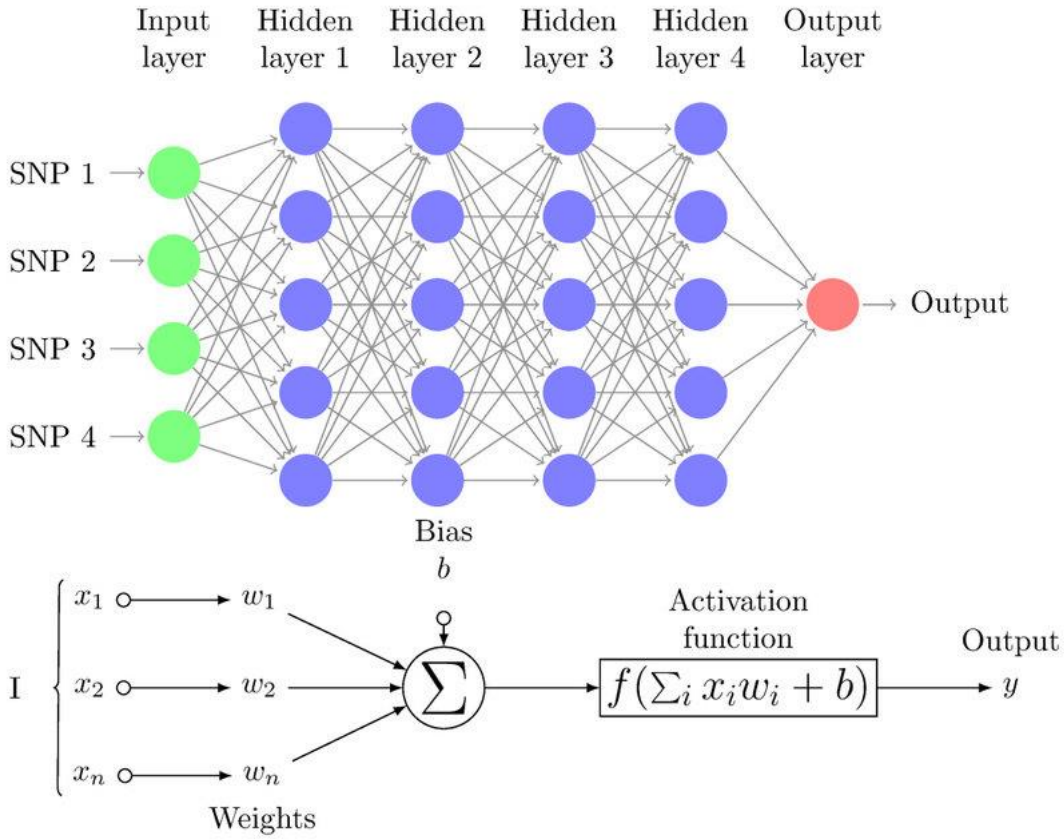


Figure 16 Multilayer perception architecture

This is a common architecture of a Multilayer Perceptron Solution, first forms the input vector and in this case represents only one document features.

Each input neuron is connected to each neuron of the next layer, these neurons shown in blue are called hidden neurons and each one of them applies an activation function, each

connection between two neurons has an associated weight represented by the letter "w" and each neuron except the input neurons has a threshold represented by the letter "b". Lastly, we see the output neuron, which in our case will return a binary value of "1", if the document is considered malicious, or "0", if the document is considered benign. There are several important measures in the architecture of this neural network:

The number of layers is known as the **depth**, and the number of units in a layer is known as the **width**. As you might guess, “deep learning” refers to training neural nets with many layers.

To better understand the functioning of Multilayer Perceptron we must explain the three phases to which this Neuron Network is exposed.

4.3.1.2. MULTILAYER PERCEPTRON PHASES

As we said previously, the input of the Multilayer Perceptron is an input vector, also called pattern, which represents the characteristics of the element to be analysed. In our model we could give an illustrative example taking into account that the objects we want to analyse are Microsoft Office documents from the e-mail. In this way, following the process that we will develop later, we would obtain a series of values associated with these documents

Table 4-1.Multilayer Perceptrons patterns

Document	**System calls number**	**String variables number**	**Int variables number**	**Operations in a row**					
Ex1.doc	0.8523	0.5634	0.7646	0.8923	…				
Ex2.doc	0.9784	0.2341	0.6453	0.1234					

The phases in which the construction of the model is broken down are Training, Validation and Testing. To carry out each of the phases of the Multilayer Perceptron we will need a set of patterns that fulfill the following characteristics:

- **Significant:** the set of patterns must have an adequate size in proportion to the complexity of the problem, reduced sets of patterns could train the model to adapt only to those patterns and reduce the capacity of generalization of the network.

- **Representative:** the set of patterns must present a similar proportion of the types to be classified, in our case it must present a proportion between benign and malicious documents in order to adapt adequately to both types.

The process of forming the sets of patterns will be developed later, showing the selection process and the tools used for this purpose with the results obtained.

4.3.1.2.1. TRAINING

The goal of this stage is to determine the value of the weights and the thresholds of the network by minimizing the error made by the network. To provide an overview of the process followed in this phase, we will show the procedure:

1. The network weights and thresholds are initialized with random values close to 0.
2. A pattern n of the training set is presented, (x(n), s(n)), and it is propagated towards the output, obtaining the result of the net y(n).

The calculation of the neuron network output follows the formula below:

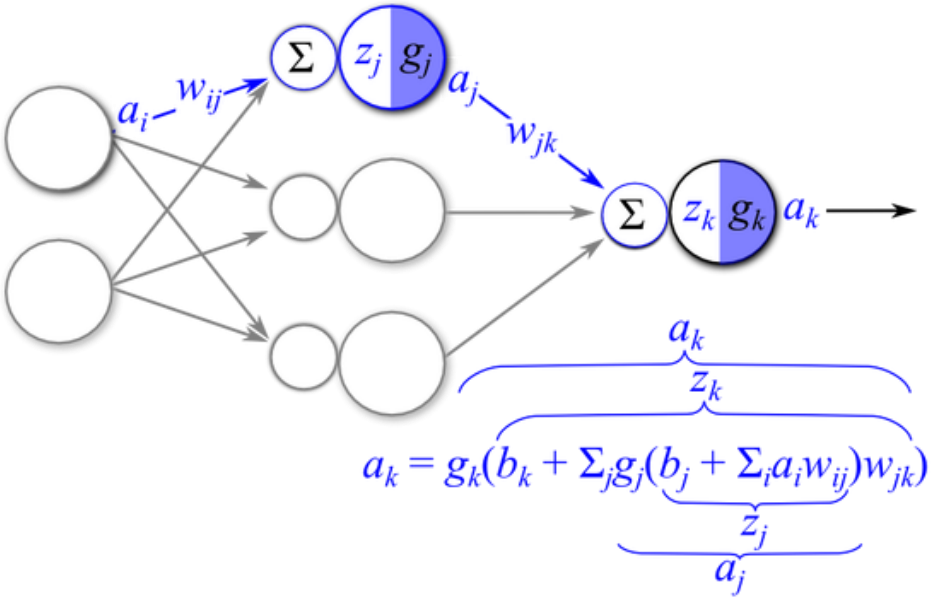


Figure 17 Forward propagate Input Signal

Where "g(n)" represents the activation function of both the hidden and the output neuron. Multilayer Perceptron normally uses non-linear activation functions to obtain curved discriminant regions, the most common activation function is Sigmoid function:

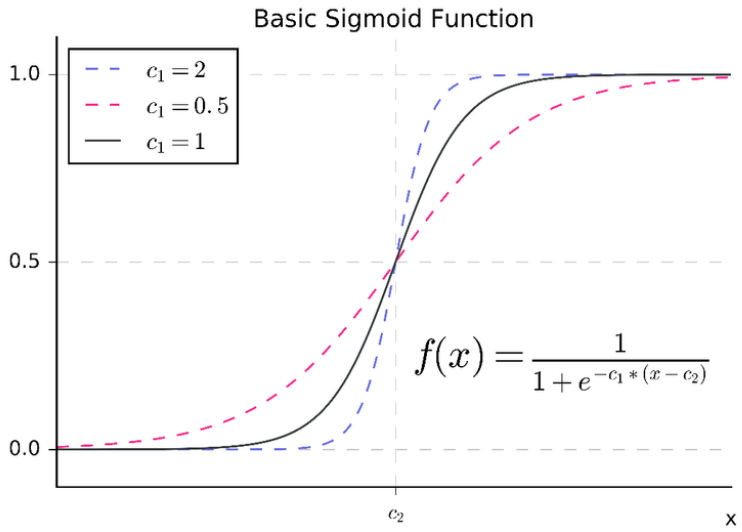


Figure 18. Basic Sigmoid Function

3. The error, e(n), made by the network is evaluated for pattern n

   The calculation of the error serves to modify the weights and thresholds proportionally to the error made by the network, in that way we will obtain significant changes when the network is far from a suitable solution and reduced changes when the network is close to the solution.

4. The generalised delta rule is applied to modify network weights and thresholds:

   4.1. The values δ are calculated for the output neurons and for the rest of the neurons in the network, starting from the output layer and back-propagating those values to the input layer.

   This process follows the following formula and procedure:

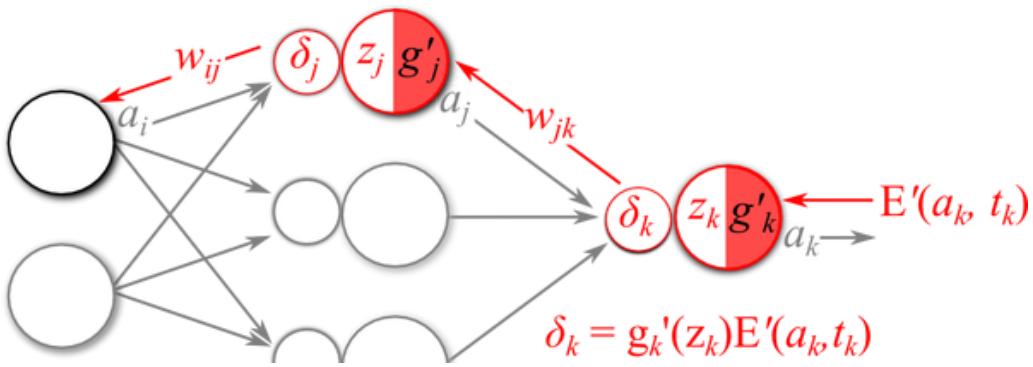


Figure 19 Back Propagate Error Signals

Thus, the δ term of an output neuron is calculated using the derivative of the activation function of that neuron and the error measured on that output neuron and the δ term of any other neuron in the network is calculated using the derivative of the activation function of that neuron and the corresponding weighted-adjusted sum of the δ terms of the neurons in the next layer. These errors are propagated backwards, reaching the first layer.

4.2. The weights and thresholds are modified:

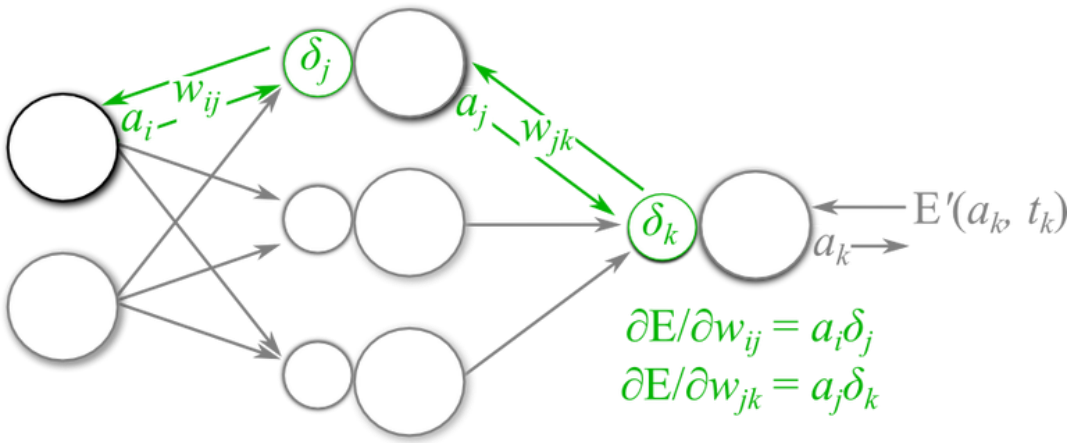


Figure 20. Weights and thresholds modification

$$w_{ij} = w_{ij} - \eta(\partial E/\partial w_{ij})$$
$$w_{jk} = w_{jk} - \eta(\partial E/\partial w_{jk})$$
for learning rate η

Figure 21. Learning rate modification

The learning rate is a value between 0 and 1 that determines the rate of change of the weights and thresholds when multiplied by the derivative of the error committed. Experimentally, the aim is to obtain the maximum learning rate that does not produce oscillations in obtaining the solution and that minimizes the number of necessary iterations.

5. Steps 2, 3 and 4 are repeated for all training patterns, completing a learning cycle.

6. The total error made by the network is evaluated (training error).

7. The validation patterns are presented, calculating only the net output, without modifying the weights.

8. The total error in the validation set is evaluated

9. Repeat steps 2, 3, 4, 5, 6 and 7 until:

    a) Training error remains stable

    b) Validation error remains stable

    c) Validation error starts to increase

4.3.1.2.2. VALIDATION

As we have seen, the Validation phase is simultaneous to the training phase and aims to compare the evolution of training results so there is no over-training of the neuron network.

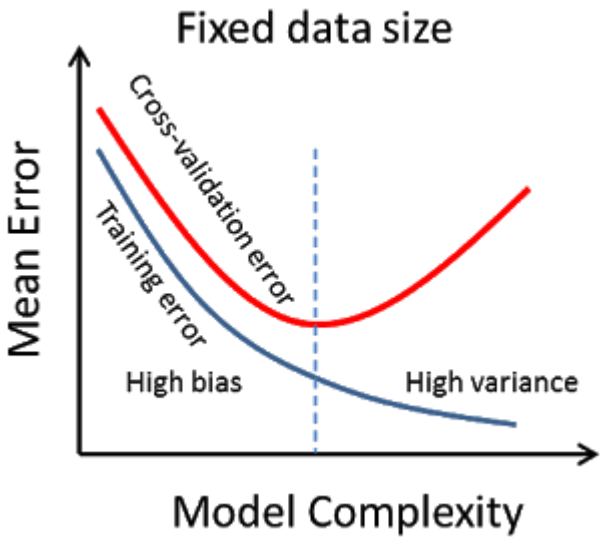


Figure 22. Optimal training

This concept can be appreciated by graphing the training error and validation error values for each iteration during the training process.

In the image below, the point marked by the dotted line represents the best fit of the training, since successively the model continues to adapt to the training set excessively and loses capacity for generalization, this phenomenon is known as overtraining.

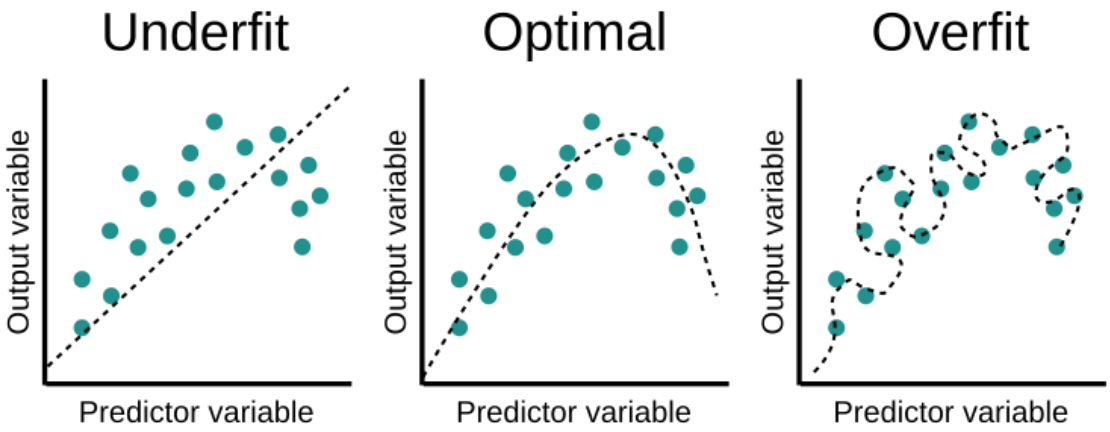


Figure 23. Tranining models

All these factors should be assessed experimentally in the training of the model to obtain an optimal final model.

4.3.1.2.3. TEST

This is the last phase to which the Multilayer Perceptron is exposed. The objective is to evaluate the generalization capacity of the final model, for which a set of patterns different to those used in the two previous phases is introduced in the network, the error obtained in the output of the set is calculated to measure the precision with which our network discriminates different patterns.

From this process we can quantify the success rate in detecting the virus in the Microsoft Office documents set that is assumed significant and representative.

4.3.2. SUPPORT VECTOR MACHINE

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that represents the number of features. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. SVM objective is to find a plane that has the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

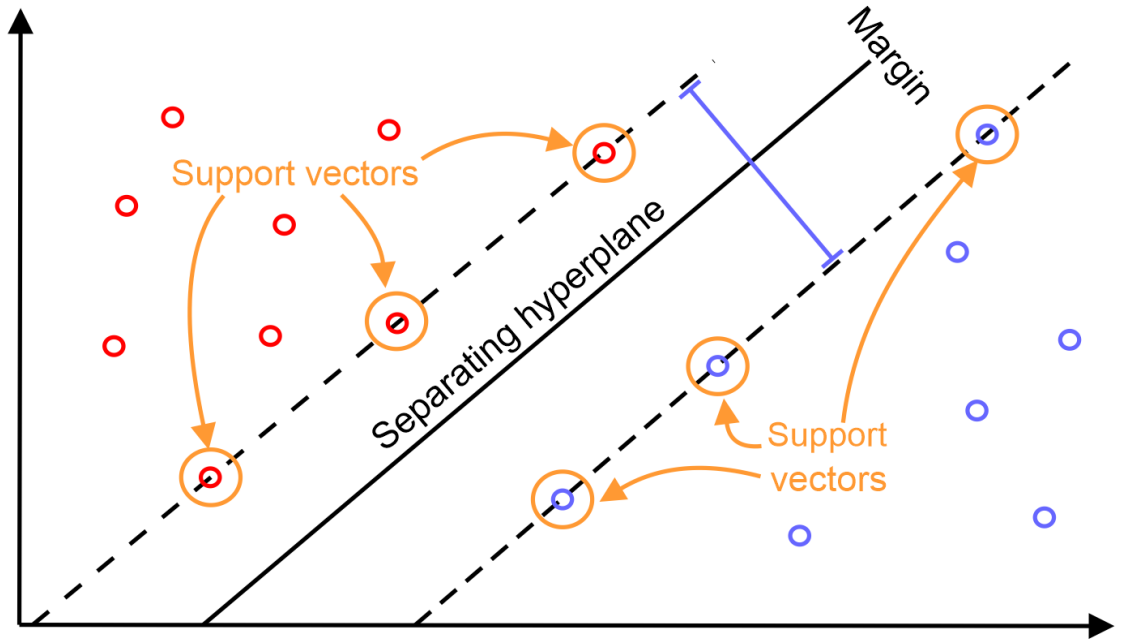


Figure 24 Optimal Hyperplane

The calculation of the optimal hyperplane is the main advantage that SVM has over MLP, among other algorithms, since respecting an equal distance between the two sets of data increases the capacity of generalization of the proposed solution.

MLP, on the contrary, could solve the previous model by proposing a hyperplane close to one of the two data sets, which would unjustifiably favour the classification of some pattern as belonging to the opposite set.

On the other hand, the calculation of the margin between the support vectors and the candidate hyperplane is obtained with quadratic propagation (QP), which implies the use of the QP matrix with a size directly proportional to the training set size.

The use of this algorithm implies a conflict between the reduction of the error in the prediction and the penalty derived from the added computational load and proportional to the number of samples in the training set.

5. PROJECT MANAGEMENT

The Project management is the application of knowledge, tools and techniques on the activities composing projects in order to collect its requirements. According to the Project Management Institute (PMI), the project management is achieved through the implementation of processes, using knowledge, abilities, tools, and management techniques.

A software development project management is responsible for ensuring the proposed solution meets end-user needs based on stakeholder design; taking into consideration the necessary resources for its development making sure they do not exceed existing resources also controlling the project evolution.

Project management processes must be well defined and known by the agents involved in the project. For this reason, there are methodologies for the project development defining these management processes, their order of realization and the resulting products.

On the other hand, it is important to conduct a pre-development study of the project to determine whether it is feasible to meet the customer's needs with the existing resources. Therefore, before determining the management processes and methodology for this task, we will carry out a feasibility study of the project.

Finally, note that the IEEE Std 830-1998 standard will be applied in the Requirements Engineering process and the methodology used to carry out the project management process will be Rational Unified Process (RUP).

5.1. FEASIBILITY STUDY

The first task on the feasibility study is to determine the functionality to attain with the project. On a commercial level, the maximum amount of information should be collected from the clients. In this case, the student and tutor should determine the scope with the desired objectives to achieve.

5.1.1. STUDY REGARDING THE CURRENT SITUATION

The main factor conditioning the feasibility of the project is the interest of the University community on the project objectives.

This project is located on computer security area, mainly related to the development of antimalware detection techniques applying Deep Learning algorithms for the detection of the Emotet virus.

Due to the specialization of the objectives and the novelty of the threat that is intended to be identified, we consider that the interest in the objectives contributes positively to the viability of the project.

On the other hand, the technological background also plays a fundamental role in the viability of this project as there are currently open source tools and specialized libraries that significantly reduce the project's workload, which also contributes positively for the consideration of viability over time, taking in to account the time to carry out the project cannot exceed the time available.

5.1.2. PEOPLE INVOLVED

There are three main agents involved in this project and are directly involved with the development of the final solution.

- **Project Jury:** The project jury. The customer will be in charge of evaluating the work comprehended in this project. The jury is not in charge of any part of the life cycle: its responsibility resides in grading the completion of this in an according manner.

- **Mentor:** The mentor oversees supervising each phase of the life cycle of the project development. The mentor (or tutor) will also have the function of simulating the client, providing additional requirements for the project. For this project, the tutor is José María De Fuentes García-Romero de Tejada.

- **Student:** The student is responsible for the completion of each phase of the system development.

5.1.3. PROJECT SCOPE

The product will consist of a Microsoft Office document analysis tool, coming mainly from the user's e-mail, being able to analyze documents from other sources such as Google Drive, the user's computer or the Internet.

The analysis of the document involves the search for evidence of the Emotet virus and uses automated analysis techniques, so that the user only has to specify the documents to be analyzed and obtain a report of this analysis from the tool, being able to download this report.

After extracting features from the macros, different classification algorithms will be used to study the malicious nature of each document that the user wants to analyze.

When developing an analysis tool for Google services: Gmail or Google Drive, the tool must maintain availability on platforms coupled with these services. The analysis must be as agile as possible in time and compatible with the greatest number of operating systems to reach the maximum number of users.

The report resulting from the tool analysis should be understandable, reducing the use of technical terminology, and will be stored along with the document macros and extracted features in a database.

It is crucial that the program runs in an environment that is isolated from the user's computer, ensuring infection prevention through the use of the tool.

5.1.4. GENERAL RESTRICTIONS

The documents to be analyzed will be Microsoft Office documents with the following extensions:".docx",".docm",".dotx",".dotm",".xlsx",".xlsm",".xltx",".xltm",".xlsb",".xlam",".pptx",".pptm",".ppsx",".ppsm",".potx",".potm",".ppam",".sldx",".sldm",".one",".mpp",".thmx"

The user must have internet access and a Google account for the use of Google services, as well as an internet connection for the communication with the database and for the download of documents to be analyzed.

5.1.5. ASSUMPTIONS AND DEPENDENCIES

The dependencies that can be identified from the description of the scope of the product are:

The use of neural models for the classification of documents will require a data set with samples of both sets to be classified and a process of training and testing of the models

With the need for isolation of the program, we can assume that we will use a hosted environment or run our program in a virtualized environment, taking into account that the program is intended for non-technical users, we must provide the greatest assistance in the process of installation and configuration of the program

5.2. SPECIFIC REQUIREMENTS

5.2.1. FUNCTIONAL REQUIREMENTS

The functional requirements definition is composed by the following elements:

- **ID:** Unique alphanumeric code for each requirement
- **Name:** Requirement title

- **Description:** Requirement explanation
- **Priority:**
    - **M:** The requirement is mandatory
    - **D:** The requirement is desirable
    - **F:** The requirement is to be postponed
    - **N:** In discussion, no longer a requirement
- **Pre_Conditions:** Conditions that need to be met to accomplish the functionality.
- **Post_Conditions:** Conditions as a result of the requirement action.
- **Non-Funtional Requirements:** Conditions for the requirement not associated to its main purpose.
- **Notes:** Any additional comment.

The functional requiremtents table is annexed at the end of the document.

5.2.2. NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements definition is composed by the following elements:

- **ID: Unique alphanumeric code for each requirement**
- **Name: Requirement title**
- **Description: Requirement explanation**
- **Notes:** Any additional comment.

The non- functional requirements table is annexed at the end of the document.

The requirements traceability matrix table is annexed at the end of the document

5.3. USE CASES

Use cases describe the exchanges between the system and the people or external systems that interact with it. Use case models are tools that contribute to the retrieval of requirements.

Use case models are composed of the use case diagram, which represent the actions that can be performed by external agents with the system, and the use case description, which shows the phases of the actions or steps needed to complete the action.

The use cases do not describe any system internal functionality nor do they explain its implementation. The use cases must have a clear interpretation and the language used will be understandable to most users.

5.3.1. AGENTS

Table 5-1. Use cases: Adminiistrator

ID	**AGT_01**
Name	Administrator
Description	Privileged agent to control, modify and monitor program status
Responsibilities	- Guarantee the end user the availability of the tool - Ensure that the program meets the established requirements - Guarantee the availability of the data stored in the application's database

Table 5-2. Use Cases: User

ID	**AGT_02**
Name	User
Description	Agent using the software tool
Responsibilities	- Use the interfaces designed for the user, taking the necessary actions for the operation of the tool - Do not perform malicious actions in the application - Respect the property rights and copyright of the tool

5.3.2. USE CASES DIAGRAMS

The language used for the construction of the use case diagrams is the Unified Model Language (UML).

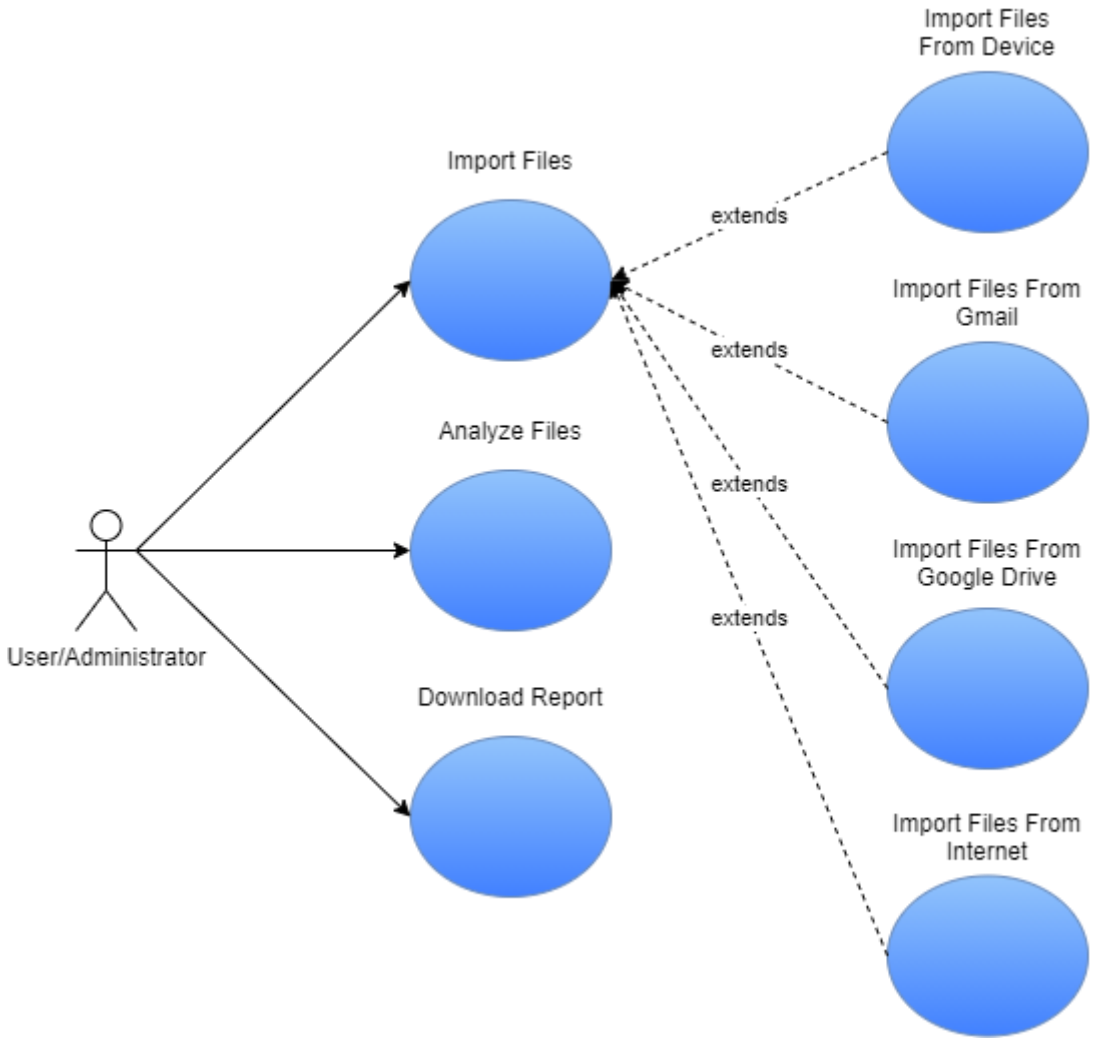


Figure 25 Use case diagram

5.3.3. USE CASES DESCRIPTION

Table 5-3 Use case descriptions

UC_ID	**UC_01**
UC_Name	Import Files From Device
Agent	AGT_01, AGT_2
Description	The use case allows the agent to import one or more Microsoft Office documents from his local device
Flow	1. Select import from local device option 2. Select the documents to import into the tool from local device
Pre-Conditions	- Have the documents for analysis on the local device
Post-Conditions	- The agent displays the imported documents in the program
Inclusion points	None
Extension points	UC_05

UC_ID	**UC_02**
UC_Name	Import Files From Gmail
Agent	AGT_01, AGT_2
Description	The use case allows the agent to import one or more Microsoft Office documents from his Gmail Account
Flow	1. Select import from Gmail Account option 2. Specify Gmail address 3. Specify Gmail password 4. Specify the extension of the documents to be imported 5. Select the documents
Pre-Conditions	- Have internet access - Have a Google Account - Have enabled the Imap option on the Google account - Have enabled the Third-party apps with account access option on the Google account
Post-Conditions	- The agent displays the imported documents in the program
Inclusion points	None

Extension points	UC_05

UC_ID	**UC_03**
UC_Name	Import Files From Google Drive
Agent	AGT_01, AGT_2
Description	The use case allows the agent to import one or more Microsoft Office documents from his Google Drive Store unit
Flow	1. Select import from Google Drive option 2. Link Google Drive account following Google Account Authentication process 3. Select the documents from Google Drive
Pre-Conditions	- Have internet access - Have a Google Account - Have enabled the Third-party apps with account access option on the Google account
Post-Conditions	- The agent displays the imported documents in the program
Inclusion points	None
Extension points	UC_05

UC_ID	**UC_04**
UC_Name	Import Files From Internet
Agent	AGT_01, AGT_2
Description	The use case allows the agent to import one or more Microsoft Office documents specifying the URL of the document
Flow	1. Select import from Internet option 2. Specify URL for file download
Pre-Conditions	- Have internet access - Have access to the database - Have a Google Account - Have enabled the Third-party apps with account access option on the Google account

Post-Conditions	- The agent displays the imported documents in the program
Inclusion points	None
Extension points	UC_05

UC_ID	**UC_05**
UC_Name	Analyze Files
Agent	AGT_01, AGT_2
Description	The use case allows the agent to perform the malware analysis over the files, obtaining a report of Emotet's malware detection
Flow	1. Select the documents for analysis 2. Run the analysis 3. Collect Emotet's malware report
Pre-Conditions	- Have internet access - Have access to the database - Have selected documents imported in the program
Post-Conditions	None
Inclusion points	None
Extension points	

UC_ID	**UC_06**
UC_Name	Download Report
Agent	AGT_01, AGT_2
Description	The use case allows the agent to perform the malware analysis over the files, obtaining a report of Emotet's malware detection
Flow	4. Select the documents for analysis 5. Run the analysis 6. Collect Emotet's malware report

Pre-Conditions	- Have internet access - Have access to the database - Have selected documents imported in the program
Post-Conditions	None
Inclusion points	None
Extension points	

5.4. ALTERNATIVE SOLUTIONS

Knowing the scope of the project and the requirements to be met, an analysis evaluating the possible technical solutions for the project development must be carried out.

We can roughly identify the following necessary factors:

- Isolated Environment
- Macros extraction tools
- Macro obfuscation
- Classification models
- Database
- Programming language

Dividing the analysis in this way, we can address the possible solutions for each factor independently, showing the pros and cons of each alternative.

5.4.1. ISOLATED ENVIRONMENT

5.4.1.1. DESKTOP APPLICATION ON VIRTUAL PLATFORM

This option consists of installing a virtualized environment and developing a desktop application isolated from the host operating system.

There are a large number of virtualization platforms on the market, based on the compilation of most of these virtualization platforms made by Wikipedia, we select as candidates the platforms that support the largest number of host operating systems, maintaining open source license.

Table 5-4. Virtualization platforms (Wikipedia, 2020)

NAME	HOST CPU	HOST OS	GUEST OS
BOCHS	Any	Windows, Linux, FreeBSD, Unix/X11, Mac OS 9, macOS, BeOS, MorphOS, OS/2	Windows, Linux, DOS, BSD, OS/2, Haiku
VIRTUAL BOX	x86, x86-64 (with Intel VT-x or AMD-V, and VirtualBox 2 or later)	Windows, Linux, macOS, Solaris, FreeBSD, eComStation	DOS, Linux, macOS, FreeBSD, Haiku, OS/2, Solaris, Syllable, Windows, and OpenBSD (with Intel VT-x or AMD-V, due to otherwise tolerated incompatibilities in the emulated memory management).
VMWARE PLAYER	x86, x86-64	Windows, Linux	Windows, Linux, Solaris, FreeBSD, OSx86 (as FreeBSD), virtual appliances, Netware, OS/2, SCO, BeOS, Haiku, Darwin, others: runs arbitrary OS
HYPER-V	x86-64 with Intel VT-x or AMD-V	Windows Server 2008 (R2) w/Hyper-V role, Microsoft Hyper-V Server	Supported drivers for Windows 2000, 2003, 2008, XP, Vista, FreeBSD, Linux (SUSE 10 released, more announced)
DOSBOX	Any	Linux, Windows, classic Mac OS, macOS, BeOS, FreeBSD, NetBSD, OpenBSD, Solaris, QNX, IRIX, MorphOS, AmigaOS, Maemo, Symbian	Internally emulated DOS shell; classic PC booter games, unofficially Windows 1.0 to 98

PROS

This solution ensures isolation from the host system during program execution, Tools such as Oracle VM Virtual Box or VMware are extensively documented and have high usability

CONS

The main drawback of this solution is the multiplicity of code for each OS to be included in the compatibility of the tool, if we want our tool to be compatible with Windows and Linux operating systems, we must branch out the development and design the tests for each Operating System.

In addition, the usability of the tool would depend on the installation of at least one of the previous virtualization platforms, environment configuration and program execution, which for a user without technology experience could be a complex process.

5.4.1.2. WEB APPLICATION

A web application is an application software that runs on a web server, accessed by the user through a web browser with an active internet connection and providing services.

This alternative consists of developing a web application, which allows users to analyze Microsoft Office documents on the Internet. This way, the application would be developed to process user requests and web server instances would be set up with the hosted program to support users.

The viable alternatives of the elements that would be necessary for the development of the web application are studied below.

5.4.1.2.1. BACKEND FRAMEWORKS

Due to the specialization of the backend framework the analysis of alternatives is a process that can provide a saving in the implementation time of the web application

Rails (RubyBochs): Great framework for metaprogramming (where a computer program can treat other programs as their data) and database orientated web programming. Small projects oriented.

Django (Python): High-level Python Web framework that encourages rapid development. Data science projects Oriented.

Laravel (PHP): Multipurpose backend framework, easy to implement MVC web applications.

5.4.1.2.2. WEB APPLICATION HOSTING SERVICE

The possible web application hosting services to be used as part of the solution are: **AWS**, **Microsoft Azure** and **Google Cloud Platform**

PROS

The advantages over the Desktop application are:

- Running the program on a web server providing isolation
- Exclusive use of the web server's computer resources
- Highly scalable model
- No need to install a virtual machine or configure the environment

CONS

The weaknesses of this solution lie in the high number of threats received by public web applications, which pose a real risk to users if they lack a cyber-security plan that can be costly.

Web applications have an associated cost of hosting on web servers and maintenance that in the long term can also mean a high monetary cost.

Finally, the development of the web application also requires the design and coding of a user interface.

5.4.1.3. CLOUD JUPYTER NOTEBOOKS

"Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document."

The solution of implementing our program using a Jupyter Notebook can be similar to using a datascience framework, but due to the popularity and reception of this project, cloud services providers such as Google and Microsoft support Jupyter Notebook online for any user.

Therefore, the solution proposed is to use a Jupyter Notebook as a service.

Table 5-5. Jupyter Notebooks Cloud Services

SERVICE	SUPPORTED LANGUAGES	FEATURES
BINDER	Python (2 and 3), R, Julia	2GB of RAM Ease of working with datasets: If your dataset is in the same Git repository
GOOGLE COLABORATORY (COLAB)	Siwft, Python (2 and 3), R, Julia	12 GB RAM GPU and TPU access Google Drive connections Hundreds of packages come pre-installed and pip service available (easy to install dependencies)
MICROSOFT AZURE NOTEBOOKS	Python (2 and 3), R, and F#	4 GB of RAM 1 GB disk per project Azure includes connectors to other Azure services, such as Azure Storage and various Azure databases.

PROS

- Multi-platform solution with a pre-configured environment and high ease of installation of dependencies.
- Running on a web server provides isolation and uses the server's computer resources.
- Does not require installation of a virtualization platform
- The possible use of GPUs contributes to the reduction of program execution time
- Eliminates the need for web application maintenance
- Eliminates the need for a security threat plan

CONS

- The execution environment may seem complex to a user without technical skills
- Implementing the tool in a single location can detract from the clarity of the code and generate coupling

5.4.2. MACRO EXTRACTION TOOLS

Table 5-6. Macro Extraction Tools

TOOLS	LANGUAGE	SUPPORTED FORMATS
OLEVBA	Python 2	Word/Excel/PowerPoint 97-2003: Yes Word/Excel/PowerPoint 2007+: Yes Word 2003 XML (.xml) Word/Excel MHTML, aka Single File Web Page (.mht)
OLEDUMP	Python 2 / Python 3	Word 97-2003: Yes Excel 97-2003: Yes PowerPoint 97-2003: No Word/Excel/PowerPoint 2007+: not directly, must extract vbaProject.bin first
OFIICE MALSCANNER	C	Word 97-2003: Yes Excel 97-2003: Yes PowerPoint 97-2003: No Word/Excel/PowerPoint 2007+: not directly.

5.4.3. MACRO OBFUSCATION

The existing tools to obfuscate macros from Github are: Ch4meleon/vba_obfuscator, Bonnet/vba_obfuscator and Pepitho/VBad.

5.4.4. CLASSIFICATION MODELS

Table 5-7. Machine Learning Frameworks

MACHINE LEARNING FRAMEWORKS	CHARACTERISTICS
TENSORFLOW	- Large Dataset - High Performance - Functionality
KERAS	- Rapid Prototyping - Small Dataset - Multiple back-end support
PYTORCH	- Flexibility - Short Training Duration - Debugging capabilities

5.4.5. DATABASE

We know from the requirements that the database will be used to store the document dataset and the analysis results. For this reason, we also know that the database does not have to maintain a relational structure, so NoSQL databases can be used.

Considering this factor and our main goal: make the program as agile as possible, we will focus on NoSQL solutions compatible with our project

Table 5-8. NoSQL Databases (Kovacs, 2020)

DATABASE	WRITTEN IN	MAIN POINT	LICENSE	PROTOCOL
MONGODB	C++	JSON document store	AGPL (Drivers: Apache)	Custom, binary (BSON)
CASSANDRA	Java	Store huge datasets in "almost" SQL	Apache	CQL3 and Thrift
REDIS	C	Blazing fast	BSD	Telnet-like, binary safe
ELASTICSEARCH	Java	Advanced Search	Apache	JSON over HTTP (Plugins: Thrift, memcached)

5.4.6. PROGRAMMING LANGUAGE

There are so many different programming languages, here we have selected two very useful for this purpose to be discussed as potential solutions.

Python: High-level, general-purpose programming language. Python is a dynamic programming language that supports object-oriented, imperative, functional, and procedural development paradigms. Python is very popular in machine learning programming.

C++: One of the oldest and most popular programming languages. Most of the machine learning platforms support C++.

5.5. PROJECT SOLUTION

Considering the benefits of computing performance, isolation in the execution, ease of development and platform compatibility offered using Google Collaboratory, this alternative becomes the most favorable alternative for the project.

Oledump is the macro extraction tool with the most updated language (Python3), for this reason, this is the tool that is selected for the solution.

With respect to the obfuscation tool, we see a difference in terms of documentation, with the Bonnet/vba_obfuscator tool being the most used and documented.

In relation to the model building frameworks we see a lot of similarity, therefore this decision is postponed at development time.

In relation to the programming language, we see that the language most compatible with the proposed tools and the most oriented to Machine Learning is Python. It is native language in the programming of Jupyter Notebooks being compatible with all the Jupyter Notebooks Cloud Services. Most of the macro extraction and obfuscation tools use this same language, which also conditions their choice. So, the use of Python as a programming language is the best option for our development.

Finally, the database selected to support the data load is MongoDB.

5.6. ESTIMATION

“Estimating is a critical part of project planning, involving a quantitative estimate of project costs, resources or duration. An estimate is frequently created by analyzing a similar project and determining if your project is a little larger or smaller than a previous project. Based on resources and the schedule used to perform the previous project, plus the difference in “perceived size,” the resources and schedule are adjusted at a macro

level to formulate a staffing plan that is used to determine the total effort and cost of a project."[PMI]

The ideal scenario would be to know the estimates of a project with similar characteristics and approximate the cost of the effort on this basis following the recommendations of the PMI. Unfortunately, we lack such an estimate, so we will make the estimation applying the technique of adjustment by points of use cases. This technique consists of obtaining an approximation of the cost in time that supposes our project starting from the use cases defined, its complexity, the actors involved, both technical and environmental factors.

A template provided by the Computer Science Department of the Universidad Carlos III de Madrid will be used for the estimation process.

5.6.1. UNADJUSTED USE CASE POINTS

5.6.1.1. USE CASE WEIGHT FACTOR

Table 5-9. Use case weight factor

Use case weight factor (Based on use case transaction number)		**Weight**	**Number**	**Weighted value**
Simple	3 or less transactions	5	5	25
Medium	4 to 7 transactions	10	3	30
Complex	more than 7 transactions	15	0	0
Unadjusted use case weight (UUCW)				**55**

5.6.1.2. ACTORS WEIGHT FACTOR

Table 5-10. Actors weight factor

Actors Weight Factor	**Description**	**Weight**	**Number**	**Weighted value**
Simple	API Program	1	0	0
Medium	Human command line or machine via protocol	2	0	0
Complex	Human with GUI	3	2	6
Unadjusted Actor Weight (UAW)				**6**

The result of unadjusted use cases is the sum of UUCW plus UAW

5.6.2. TECHNICAL WEIGHT FACTOR

Table 5-11. Technical weight factor

Technical Weight Factors	Assignment Scale	Weight	Number	Weighted value
T1 Distributed System	0=not important 5=essential	2	0	0
T2 Performance Objectives or Response Time	0=not important 5=essential	1	2	2
T3 End User Efficiency	0=not important 5=essential	1	2	2
T4 Complex Internal Processing	0=not important 5=essential	1	4	4
T5 Code Must Be Reusable	0=not important 5=essential	1	3	3
T6 Ease of Installation	0=not important 5=essential	0,5	0	0
T7 Ease of Use	0=not important 5=essential	0,5	3	1,5
T8 Portability	0=not important 5=essential	2	0	0
T9 Ease of Change	0=not important 5=essential	1	3	3
T10 Concurrency	0=not important 5=essential	1	0	0
T11 Includes Special Safety Features	0=not important 5=essential	1	0	0
T12 Provides Direct Access to Third Parties	0=not important 5=essential	1	5	5
T13 Special User Training Aids Required	0=not important 5=essential	1	4	4
Technical Factors				**24,5**
Technical Complexity Factor (TCF)	**.06 + (.01*Technical Factors)**			**0,845**

5.6.3. ENVIRONMENTAL WEIGHT FACTOR

Table 5-12. Environmental weight factor

Environmental Weight Factors	0 to 5 Scale	Weight	Number	Weighted value
F1 Familiarity with a Defined Process (LARMAN, RUP, etc.)	0 = no experience, 3=average, 5=expert	1,5	4	6
F2 Experience in the Application Domain	0 = no experience, 3=average, 5=expert	0,5	4	2
F3 Experience in Object Orientation	0 = no experience, 3=average, 5=expert	1	4	4
F4 Analyst Leadership Capacity	0 = no experience, 3=average, 5=expert	0,5	4	2

Environmental Weight Factors	0 to 5 Scale	Weight	Number	Weighted value
F5 Motivation	0=no, 3=medium, 5=high	1	5	5
F6 Stable Requirements	0=extremely unstable, 5=no change	2	5	10
F7 Part-Time Members	0=full time 5=part time	-1	0	0
F8 Programming Language Difficulty	0=easy, 3=medium, 5=hard	-1	1	-1
Environmental Factors				**28**
Environmental Factors (EF)	**1.4 + (-0.03*Environmental Factors)**			**0,56**

5.6.4. ADJUSTED USE CASE POINTS

Table 5-13. Adjusted case points

Adjusted Use Case Point (UCP)		28,8652	
Effort Hours Person per Point of Use Case		20	hours.use-case
Effort Person Hours Initial estimate (coding)		577,304	hours.man
Effort Hours Estimated Person in Project		1443,26	hours.man
Effort Months Person Estimated in the Project		9,020375	Months
Hours of work per month	**160**		

5.6.5. ACTIVITY HOURS

Table 5-14. Activity hours

Activity	Percentage	Hours.man
Analysis	10%	144,33
Design	20%	288,65
Programming	40%	577,30
Tests	15%	216,49
Overload(other activities)	15%	216,49

5.7. PLANNING

Table 5-15. Task planner

TASK NAME	DURATION	START	END	PREDECESSOR
PHASE 0: PRELIMINARY	**126 hrs**	**mié 08/01/20**	**mié 29/01/20**	
PHASE 1: PLANNING AND REQUIREMENT ESPECIFICATION	**64 hrs**	**mié 29/01/20**	**lun 10/02/20**	**14**
PHASE 2: CONSTRUCTION	**1170 hrs**	**lun 10/02/20**	**mar 01/09/20**	**24**
ITERATION 1	**1170 hrs**	**lun 10/02/20**	**mar 01/09/20**	
ANALYSIS	**14 hrs**	**lun 10/02/20**	**mié 12/02/20**	
Use Case Detailed Description	12 hrs	lun 10/02/20	mié 12/02/20	23
Use Case Detailed Description Review	2 hrs	mié 12/02/20	mié 12/02/20	28
DESIGN	**46 hrs**	**mié 12/02/20**	**jue 20/02/20**	
Sequence Diagrams	24 hrs	mié 12/02/20	lun 17/02/20	29
Sequence Diagrams Review	4 hrs	lun 17/02/20	lun 17/02/20	31
Class Diagrams	14 hrs	mar 18/02/20	mié 19/02/20	32
Class Diagrams Review	4 hrs	mié 19/02/20	jue 20/02/20	33
Design Initialization Milestone	0 hrs	jue 20/02/20	jue 20/02/20	34
CODING	**800 hrs**	**jue 20/02/20**	**jue 09/07/20**	**35**
Acquainting the environment	40 hrs	jue 20/02/20	jue 27/02/20	35
Dependencies study	20 hrs	jue 27/02/20	lun 02/03/20	37
Database creation and administration	40 hrs	lun 02/03/20	lun 09/03/20	38
Upload from local device	15 hrs	lun 09/03/20	mié 11/03/20	39
Link Google Drive Account	25 hrs	mié 11/03/20	lun 16/03/20	40
Import from Gmail	30 hrs	lun 16/03/20	vie 20/03/20	41

TASK NAME	DURATION	START	END	PREDECESSOR
Upload from Internet option	15 hrs	vie 20/03/20	mar 24/03/20	42
Database access configuration	55 hrs	mar 24/03/20	jue 02/04/20	43
Macro Extraction	30 hrs	jue 02/04/20	mar 07/04/20	44
Features Extraction	150 hrs	mié 08/04/20	lun 04/05/20	45
Calculate Euclidean distance	100 hrs	lun 04/05/20	jue 21/05/20	46
Create Tfidf Matrix	35 hrs	jue 21/05/20	mié 27/05/20	47
Calculate Code Similarity	100 hrs	mié 27/05/20	lun 15/06/20	48
Processing data for Model Training	35 hrs	lun 15/06/20	vie 19/06/20	49
Building Neural Network	30 hrs	vie 19/06/20	jue 25/06/20	50
Training Neural Network	45 hrs	jue 25/06/20	jue 02/07/20	51
Training Results Study	20 hrs	jue 02/07/20	mar 07/07/20	52
Model Training Updates and Improvements	15 hrs	mar 07/07/20	jue 09/07/20	53
Coding End Milestone	0 hrs	jue 09/07/20	jue 09/07/20	54
TESTING	**310 hrs**	**jue 09/07/20**	**mar 01/09/20**	**55**
Import document testing	10 hrs	jue 09/07/20	vie 10/07/20	55
Database Testing	15 hrs	vie 10/07/20	mar 14/07/20	57
SVM Model Taining Testing	60 hrs	mar 14/07/20	jue 23/07/20	58
MLP Model Training Testing	60 hrs	jue 23/07/20	mar 04/08/20	59
Features Extraction Testing	150 hrs	mar 04/08/20	lun 31/08/20	60
Obfusction Testing	15 hrs	lun 31/08/20	mar 01/09/20	61
ITERATION 1 Milestone	0 hrs	mar 01/09/20	mar 01/09/20	62
PHASE 2 Milestone	0 hrs	mar 01/09/20	mar 01/09/20	63

The Gantt Diagram is annexed at the end of the document.

5.8. BUDGET

It is essential when carrying out a project to have the budget that would entail develop it. For this reason, in this section we will analyze the costs of personnel and software and hardware used for the development of the project in order to obtain an estimated Budget if it had to be presented to a company. The costs incurred are:

EMPLOYEES

The vast majority of this project has been carried out by one person, who has acted as head of product, designer and developer. The average salary of these profiles can be established at 30,000 euros gross per year, being therefore the cost per hour for the company after taxes of 22 euros per hour.

The duration of the project as shown in the previous planning was 1360 hour the cost of staff has been 1500* 22 =33000 euros

Table 5-16. Budget: Staff

NAME	POSITION	WAGE	TOTAL HOURS	COST
MIGUEL PEIDRO	Developer, designer	22 €/h	1500	29.920 €
JOSÉ Mª DE FUENTES	Supervisor	35€/h	150	5.250 €
TOTAL				35.170€

HARDWARE

The hardware elements used for the realization of the project are:

1. Laptop amortized in 2 years (24 months):

Lenovo IdeaPad S540-14IML- Processor: Intel Core i7- 10510U

- RAM: 8.00 GB DDR4

- Storage: 1TB SSD

- Operating system: Windows 10 Home

- Screen size: 14".

- Cache memory: 8MB

- Processor speed: 1,8GHz

Price: 850 Euros

To obtain the cost of use, the cost of amortization will be calculated taking into account that the project has lasted 9 months.

Table 5-17 Budget: Hardware

PRODUCT	PRICE	USE	USEFUL LIFE	COST
LAPTOP	850,00 €	9 months	24 months	318.75[2]

SOFTWARE

Several licensed programs were necessary for the development of this Project. Listed below:

1. Microsoft Project Standard 2019: 849 euros[3]. Amortized in 5 years (60 months)
2. Google Colaboratory Pro: 9.99 € per month[4]
3. Microsoft 365 Personal: 69 € per year[5]
4. MongoDB Atlas Multi-region Cluster: 95 $ per month[6]

Table 5-18 Budget: Software

PRODUCT	PRICE	USE	USEFUL LIFE	COST
PROJECT 2019	849 €	9 months	60 months	127,5 €

[2] Price retrieved from Media-Markt [https://www.mediamarkt.es/es/product/_port%C3%A1til-lenovo-ideapad-s540-14iml-14-full-hd-intel%C2%AE-core%E2%84%A2-i7-10510u-8gb-1tb-ssd-windows-10-home-gris-1475622.html]
[3] Price retrieved from Microsoft store: https://www.microsoft.com/es-es/microsoft-365/project/compare-microsoft-project-management-software
[4] Price retrieved from Google Colab Pro: https://colab.research.google.com/signup
[5] Price retrieved from Microsoft store https://www.microsoft.com/es-es/microsoft-365/buy/compare-all-microsoft-365-products
[6] Price retrieved from MongoDB https://www.mongodb.com/pricing

PRODUCT	PRICE	USE	USEFUL LIFE	COST
GOOGLE COLAB PRO	9,99 €	9 months		89,91 €
MICROSOFT 365	69 €	9 months	12 months	51,75 €
MONGODB ATLAS	95 €	9 months		855 €
TOTAL				1124,16 €

TOTAL

Once all the costs of the project have been found separately, a total estimate of the project's budget.

Table 5-19 Budget: Total

CONCEPT	PRICE
STAFF	35.170 €
HARDWARE	850 €
SOFTWARE	1.124,16 €
TOTAL	37.144.16 €

5.9. SOFTWARE CONFIGURATION MANAGEMENT

This section describes the software configuration management activities that should be carried out during the project development process.

5.9.1. SCM SCOPE

Products to be put under configuration control and procedures to be followed during the project development process are defined. Tasks such as ensuring the status of configuration items and project documentation are responsibilities that are also defined in the software configuration management process.

The changes in the development process of a project are critical events. It is necessary to evaluate the change and the way to approach it before any modification. During the change is needed to know the dependencies of the modified elements to avoid discrepancies.

5.9.2. RESPONSIBILITIES

The configuration management process involves the following responsibilities:

- **Change Control Board (CCB):** Judging if a change should be made and how it should be prioritized
- **Software Configuration Management Responsible (SCMR):** The SCMR must provide the infrastructure and configuration environment for the project. It must be concerned with all members of the group understand and are able to execute the MCS activities assigned, as well how to ensure that these are carried out. SCMR must also follow the baseline, controlling versions and changes.
- **Librarian:** Save the project's products and maintain their integrity
- **Developers:** Follow the guidelines of the configuration management process

These responsibilities will fall on the student, except for the Librarian's responsibility, which will be carried out by the Universidad Carlos III de Madrid, preserving this document, and maintaining its integrity.

5.9.3. CONFIGURATION ELEMENTS

The format to identify the configuration elements:

EC X.Y, where X determines the realization phase and Y is a numerical identification of the element.

Table 5-20. Configuration elements

Identifier	Definition	State	Date
EC 1.1	• Use case diagrams	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 1.2	• Requirements	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 1.3	• System Feasibility	• Acepted	• 27/04/2020

Identifier	Definition	State	Date

EC 2.4	• Configuration Management	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 2.5	• Quality Management	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 2.6	• Planning	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 2.7	• Estimation	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 3.8	• Program Code	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 3.9	• Code Dependencies	• Acepted	• 27/04/2020

Identifier	Definition	State	Date
EC 4.10	• Tests	• Acepted	• 27/04/2020

5.9.4. RELATIONS BETWEEN ELEMENTS

Table 5-21. Relations between elements

EC 1	EC 2	Relationship Type	Date
• **EC 1.3**	• **EC 1.1**	• Dependency	• 27/04/2020

EC 1	EC 2	Relationship Type	Date
• **EC 1.3**	• **EC 1.2**	• Dependency	• 27/04/2020

EC 1	EC 2	Relationship Type	Date
• **EC 3.8**	• **EC 3.9**	• Dependency	• 27/04/2020
EC 1	**EC 2**	**Relationship Type**	**Date**
• **EC 3.8**	• **EC 3.10**	• Dependency	• 27/04/2020

5.9.5. BASELINES

A baseline is defined as a point in the software life cycle where configuration control is applied to several specific configuration elements. The objective of a baseline is to reduce a project's vulnerability to uncontrolled change by fixing and formally change controlling various key deliverables at critical points in the development evolution.

Define the baseline with the following fields:

- **Baseline Identifier:** The identifier format used is LB X, where X is a numerical identifier.
- **EC Identifier:** The configuration elements identifier which constitute the baseline.
- **Deadline:** The point in the project where the baseline is expected to be complete.
- **Approval Method:** The method to be adopted to approve baseline.

Table 5-22. Baselines

Baseline Identifier	EC Identifier	Deadline	Approval Method
• **LB 1**	• **EC 1.1** • **EC 1.2**	• Feasable Study end	Review requirements and use cases

Baseline Identifier	EC Identifier	Deadline	Approval Method
• **LB 2**	• **EC 1.1** • **EC 1.2** • **EC 2.5** • **EC 2.6** • **EC 2.7**	Analysis phase end	Review requirements and use cases, quality plan, planning and estimation

Baseline Identifier	EC Identifier	Deadline	Approval Method
• **LB 3**	• **EC 1.1** • **EC 1.2** • **EC 3.8** • **EC 3.9** • **EC 4.10**	Testing phase end	Review requirements and use cases comparing with implementation

6. ANALYSIS AND DESIGN

The analysis process allows us to specify what the system should do. The requirements analysis will consist of a high-level description of the program behaviour.

To further explore system performance the sequence diagrams will be built. These diagrams describe the interaction of the actors with the system, showing the flow of information and the necessary actions to be taken.

The objective of this process is to set the operations of the system and we must remain open to possible modifications of the requirements.

6.1. ANALYSIS

6.1.1. SEQUENCE DIAGRAMS

As the sequence diagrams are based on the interaction of the actors, we will take as a starting point the use cases of the system.

6.1.1.1. IMPORT FORM INTERNET

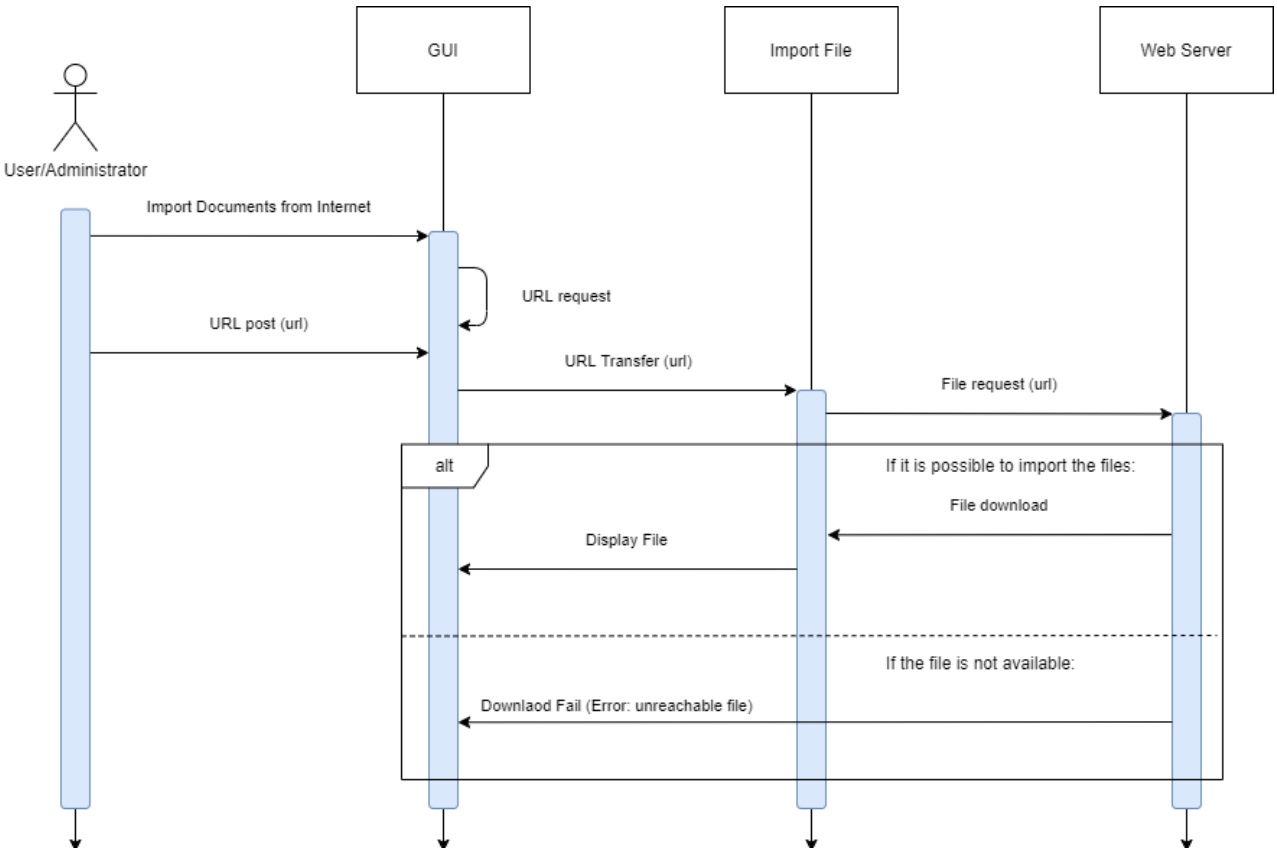


Figure 26. Import from Internet Sequence Diagram

6.1.1.2. IMPORT FROM GMAIL

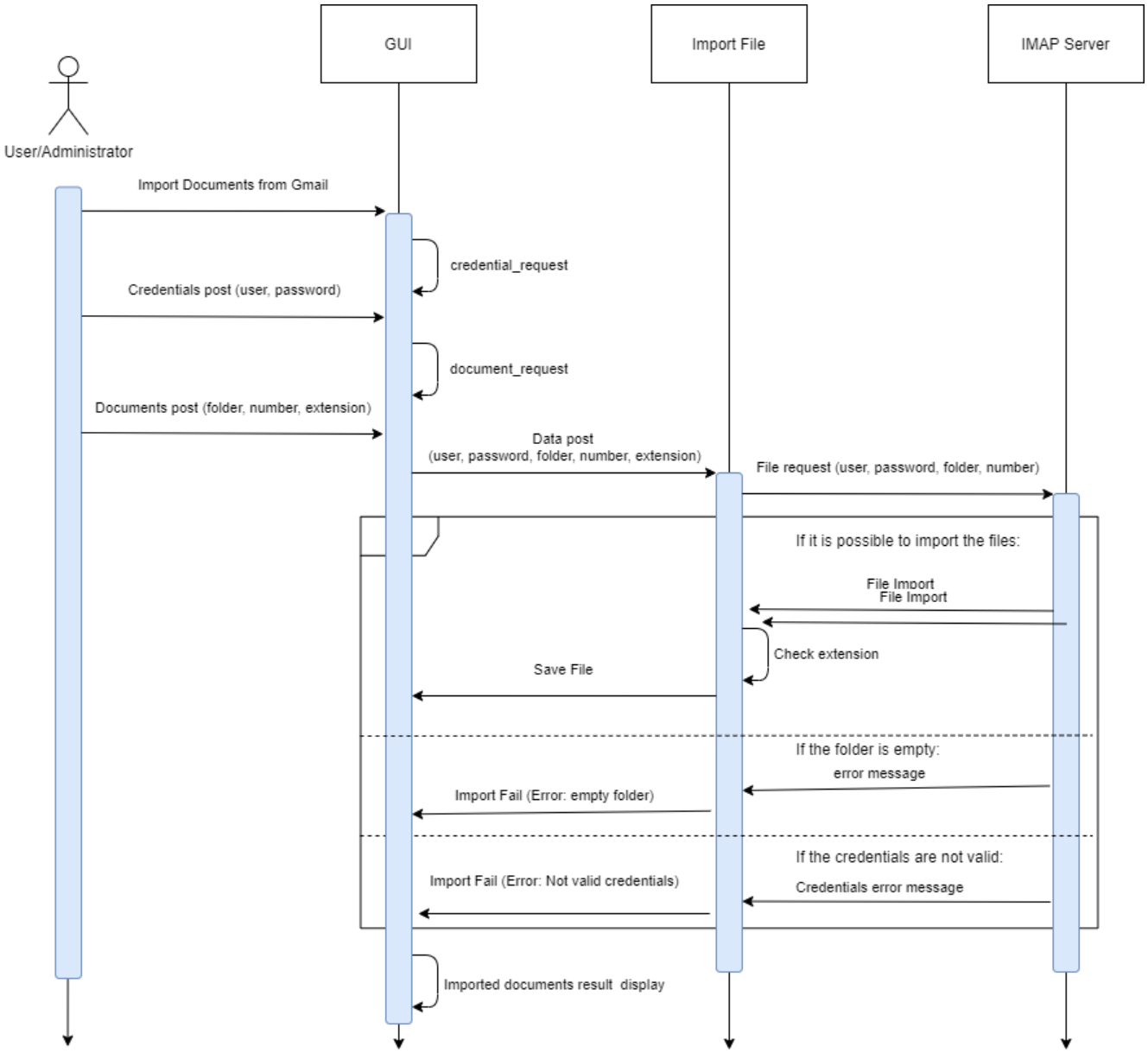


Figure 27. Import from Gmail Sequence Diagram

6.1.1.3. DOCUMENT ANALYSIS

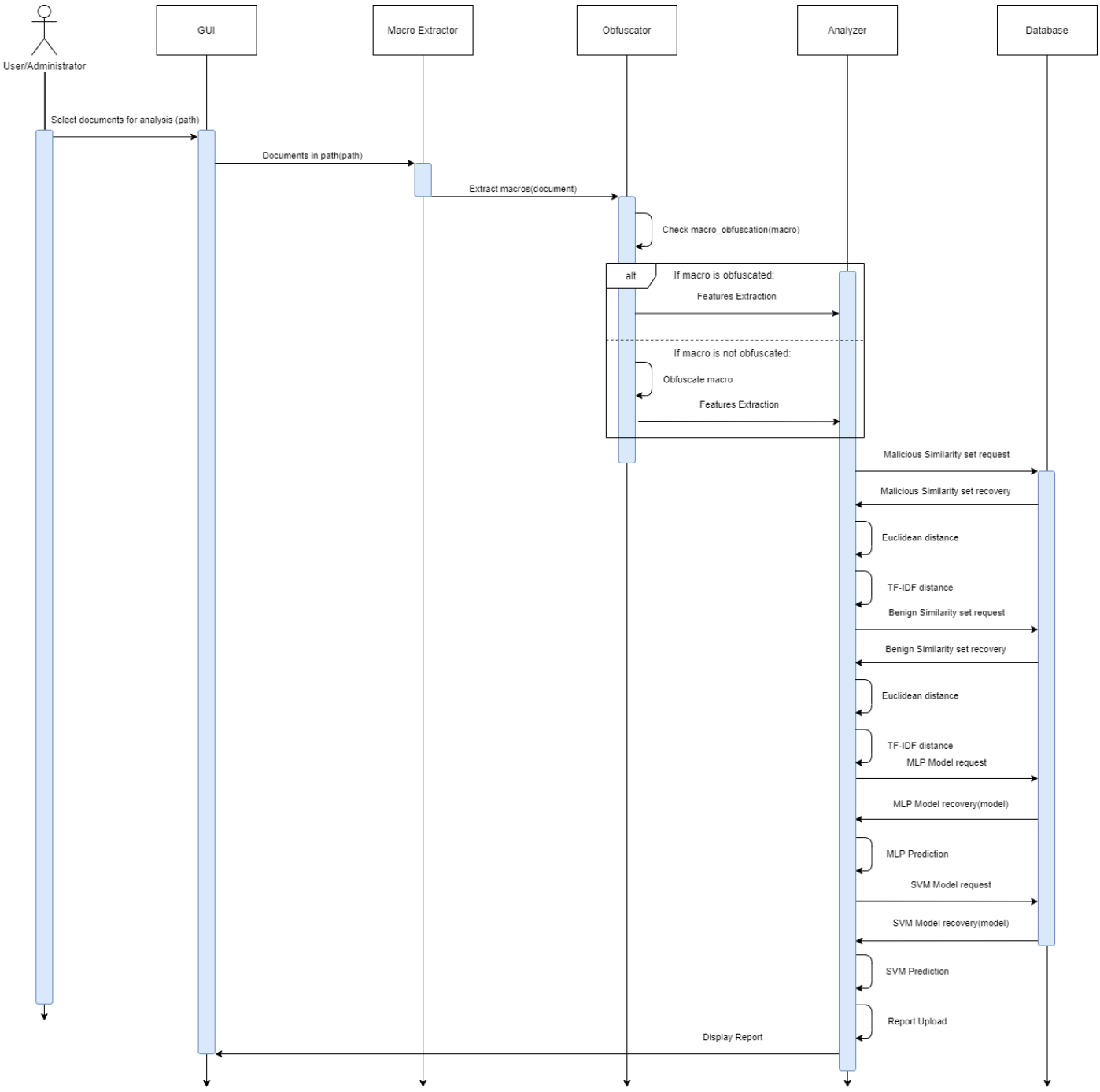


Figure 28. Analysis Sequence Diagram

6.2. DESIGN

The design process is responsible for defining the structure of the complete system, the objective is to show the construction of the system detailing the technical aspects of its development from abstractions and representations.

To perform this design task, we will use the 4+1 model, which describes the system architecture using several concurrent and specialized views and provides a complete vision of the software to be developed.

The views of the 4+1 model include the following:

6.2.1. LOGICAL VIEW

The logical view is concerned with the functionality that the system provides to end-users. (Architectural Blueprints—The “4+1” View, 1995)

As we have mentioned before, the data structure required by the program does not need a relational scheme, for this reason we have selected the MongoDB database for the realization of the project.

In this section we show the collections needed for the tool's operation:

Malicious_similarity_dataset: This collection is required for the calculation of the similarity between the macro to be analyzed with the macros extracted from the malicious samples. It will therefore consist of the malicious macros resulting from the obfuscation process.

Benign_similarity_dataset: This collection is needed for the calculation of the similarity of the macro to be analyzed with the macros extracted from the benign samples. It will therefore be composed of the benign macros resulting from the of the obfuscation process.

Training_set: This dataset is composed of the values obtained from the extraction of characteristics from each document. The objective is the creation of a data set for the machine learning models training. It is required to store the features values and the malicious nature of the document in numerical format: "1" for malicious documents and "0" for benign documents.

Reports: In this collection the results of the documents analysis will be stored.

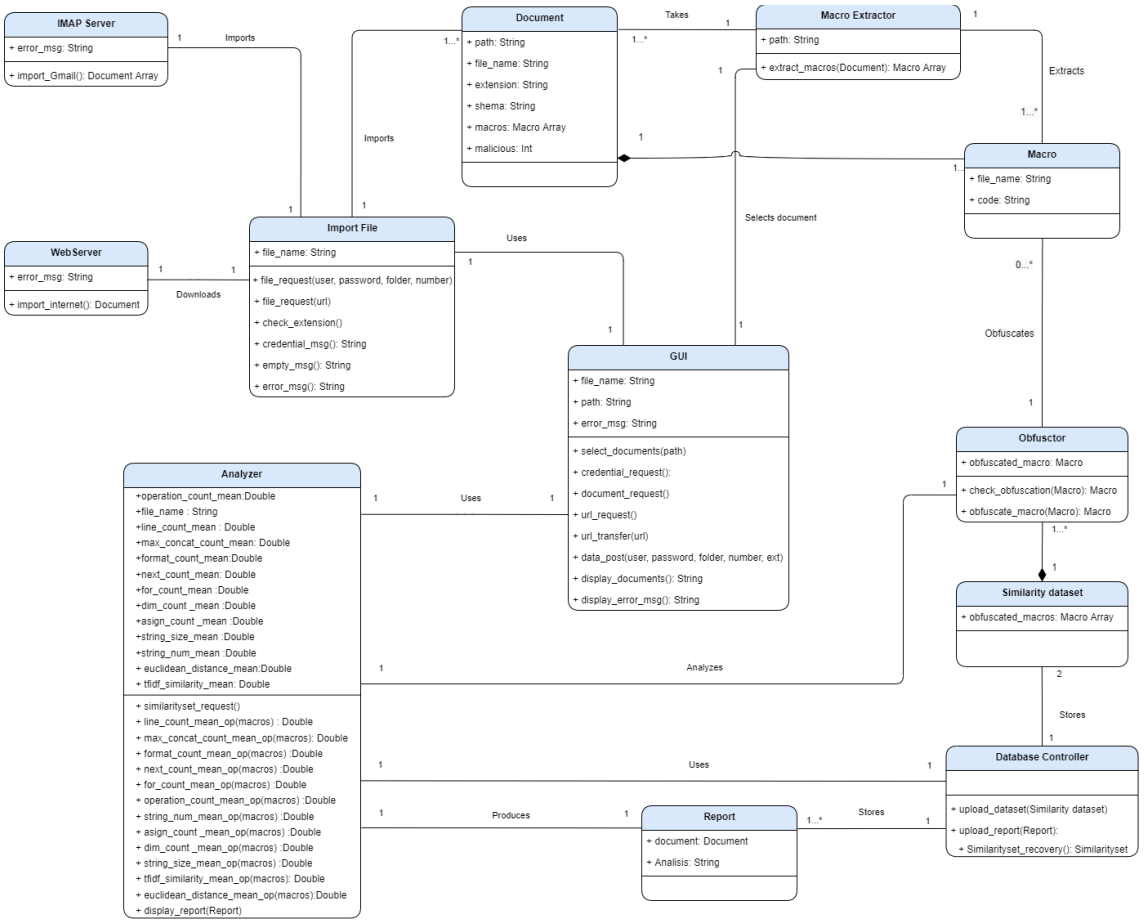


Figure 29. Class Diagram

6.2.2. DEVELOPMENT VIEW

The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML Component diagram to describe system components. (Kruchten, 1995)

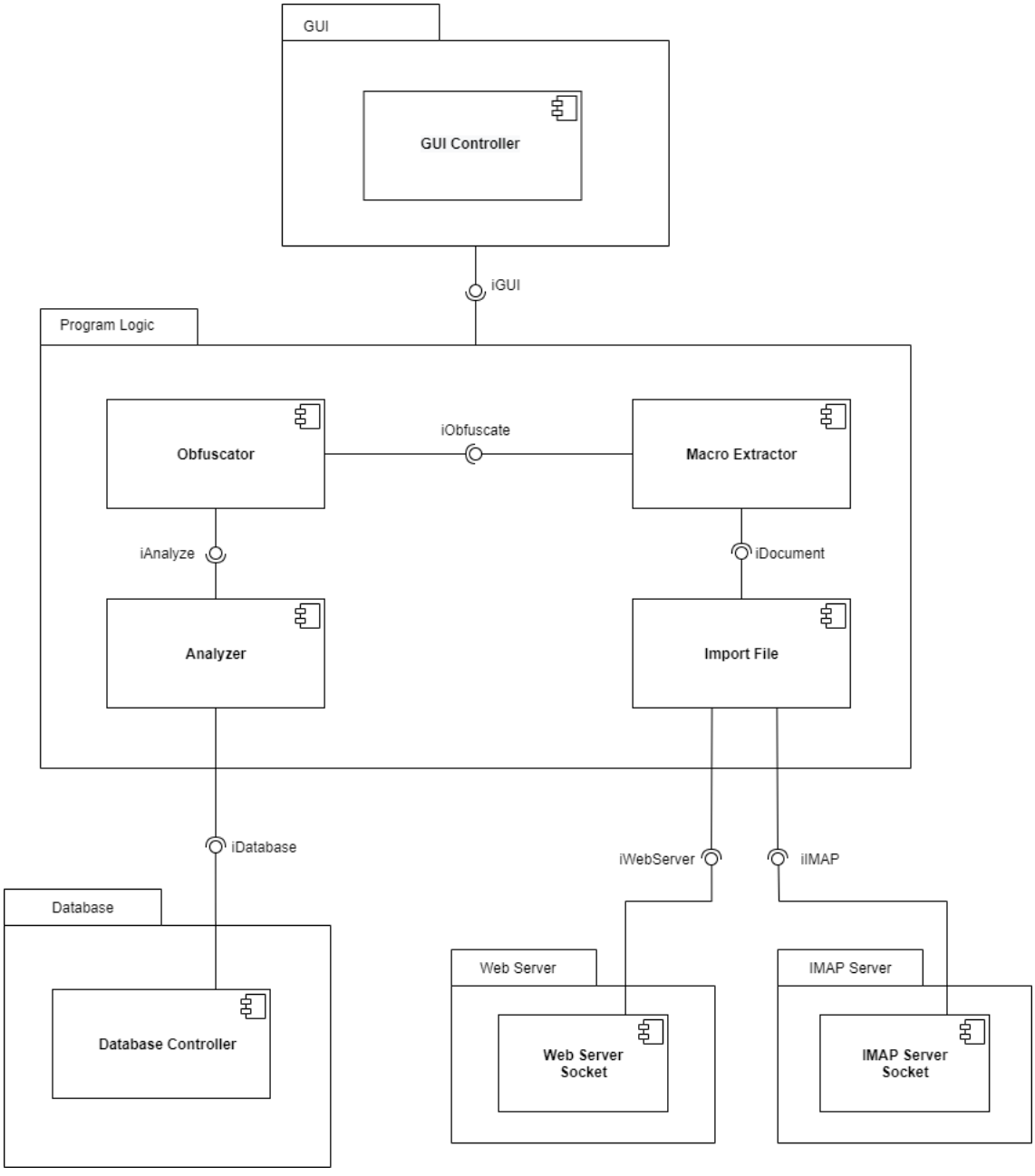


Figure 30 Components Diagram

Table 6-1 Operations Contracts: GUI Controller

GUI Controller		
Purpose	This component allows users to communicate with the system and know their status at any time.	
Dependencies		
Interfaces	• iGUI	
Operation Contracts	**select_documents (path)**	
	Description	This method extracts documents from folders and provide these documents to Macro Extractor Module
	Precondition	• Imported documents
	Postcondition	• Documents for analysis will be displayed
	credential_request ()	
	Description	Request Gmail address and password to the user
	Precondition	• Existing Gmail account with IMAP enabled and third party enabled.
	Postcondition	• Gmail account login
	document_request ()	
	Description	Request import folder, document extension and number of documents to import
	Precondition	• Logged Gmail account.
	Postcondition	• Documents will be imported
	url_request ()	
	Description	Request url for file download

GUI Controller		
	Precondition	• Existing url
	Postcondition	• Downloaded file will be dispayed
	url_transfer (url)	
	Description	Send url to File Import Module to download the file
	Precondition	• Existing url
	Postcondition	• Downloaded file will be dispayed
	data_post (user, password, folder, extension)	
	Description	Send necessary data to Import File Module to import from Gmail
	Postcondition	• Imported documents will be displayed
	display_documents()	
	Description	Show documents in session
	Precondition	• Imported document
	Postcondition	• file_name will be displayed
	display_err_msg()	
	Description	Show error during documents import
	Postcondition	• error information will be displayed

Table 6-2. Operations Contracts: IMAP Server Socket

IMAP Server Socket		
Purpose	This component allows program to import files from Gmail.	
Dependencies		
Interfaces	• iIMAP	
Operation Contracts	**Import_Gmail ()**	
	Description	This method sends the document requested to the program
	Precondition	• Document requested
	Postcondition	• Document requested will be sent

Table 6-3. Operations Contracts: Web Server Socket

Web Server Socket		
Purpose	This component allows program to download files.	
Dependencies		
Interfaces	• iWebServer	
Operation Contracts	**Import_internet ()**	
	Description	This method downloads file requested
	Precondition	• Document requested
	Postcondition	• Document requested will be downloaded

Table 6-4. Operation Contracts: Import File

Import File		
Purpose	This component allows users to import files from different sources	
Dependencies	• iGUI, iIMAP, iWebServer	
Interfaces	• iDocument	
Operation Contracts	**File_request (user, password, folder, number)**	
	Description	This method allows the user to specify values and request files from Gmail
	Precondition	• Existing Gmail account with IMAP enabled and third party enabled.
	Postcondition	• Documents imported will be displayed
	Credential_msg ()	
	Description	Display credential error
	Empty_msg ()	
	Description	Display empty message error
	Error_msg ()	
	Description	Display error message

Table 6-5. Operation Contracts: Macro Extractor

Macro Extractor	
Purpose	This component allows program to extract documents macros.
Dependencies	• iDocument

Macro Extractor		
Interfaces	• iObfuscated	
Operation Contracts	**Extract_macros (Document)**	
	Description	This method extracts the macros of the document selected
	Postcondition	• Macro content will be stored in txt file

Table 6-6. Operation Contracts: Obfuscator

Obfuscator		
Purpose	This component allows program to obfuscate macros.	
Dependencies	• iObfuscated	
Interfaces	• iAnalyze	
Operation Contracts	**Check_obfuscation (Macro)**	
	Description	This method determines if the macro is obfuscated
	Precondition	• Macro content will be stored in txt file
	Obfuscate_macros (Macro)	
	Description	This method obfuscates macros
	Precondition	• Macro content will be stored in txt file

Table 6-7. Operation Contracts: Analyzer

Analyzer		
Purpose	This component extract features from obfuscated macros	
Dependencies	• iGUI, iDatabase, iAnalyze	
Interfaces		
Operation Contracts	**Similarityset_request (path)**	
	Description	This method requests a similarity set to perform similarity calculation
	Precondition	• Similrityset in database
	Line_count_mean ()	
	Description	Calculates mean value of lines in document macros
	Precondition	• Obfuscated macros
	Max_concat_count_mean ()	
	Description	Calculates mean value of max operation in a row
	Precondition	• Obfuscated macros
	Operation_count_mean ()	
	Description	Calculates mean value of operations in document macros
	Precondition	• Obfuscated macros
	String_num_mean ()	
	Description	Calculates mean value of strings in document macros

Analyzer		
	Precondition	• Obfuscated macros
	String_size_mean ()	
	Description	Calculates mean value of strings size in document macros
	Precondition	• Obfuscated macros
	Asign_count_mean()	
	Description	Calculates mean value of asignations in document macros
	Precondition	• Obfuscated macros
	Dim_count_mean()	
	Description	Calculates mean value of Dim statements in document macros
	Precondition	• Obfuscated macros
	for_count_mean()	
	Description	Calculates mean value of for loops in document macros
	Precondition	• Obfuscated macros
	Tfidf_similarity_mean()	
	Description	Calculates the tfidf matrix for document macros
	Precondition	• Obfuscated macros
	Euclidean_distance_mean()	
	Description	Calculates the Euclidean distance in document macros
	Precondition	• Obfuscated macros

Analyzer		
	Display_report()	
	Description	Displays the report with the results
	Precondition	• Obfuscated macros

Database Controller		
Purpose	This component allows program to use the Data Base.	
Dependencies		
Interfaces	• iDatabase	
Operation Contracts	**Upload_dataset()**	
	Description	This method saves the data set on the Data Base
	Precondition	• Data Base Access • Data Set to save
	Postcondition	• Data set will be saved
	Upload_report()	
	Description	This method saves the report on the Data Base
	Precondition	• Data Base Access • Report to save
	Postcondition	• Report will be saved
	Similarityset_recovery()	
	Description	This method imports the data set from the Data Base
	Precondition	• Data Base Access

Database Controller		
		• Data Set to import
	Postcondition	• Data set will be imported

6.2.3. PROCESS VIEW

The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system. (Kruchten, 1995)

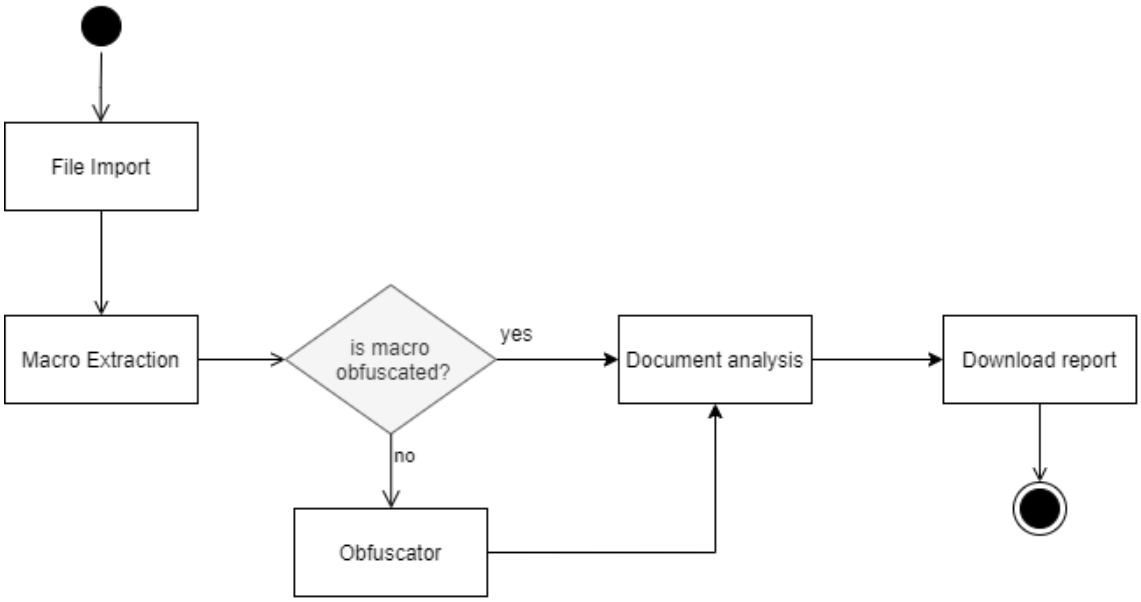


Figure 31 Process diagram

6.2.4. PHYSICAL VIEW

The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer as well as the physical connections between these components. (Kruchten, 1995)

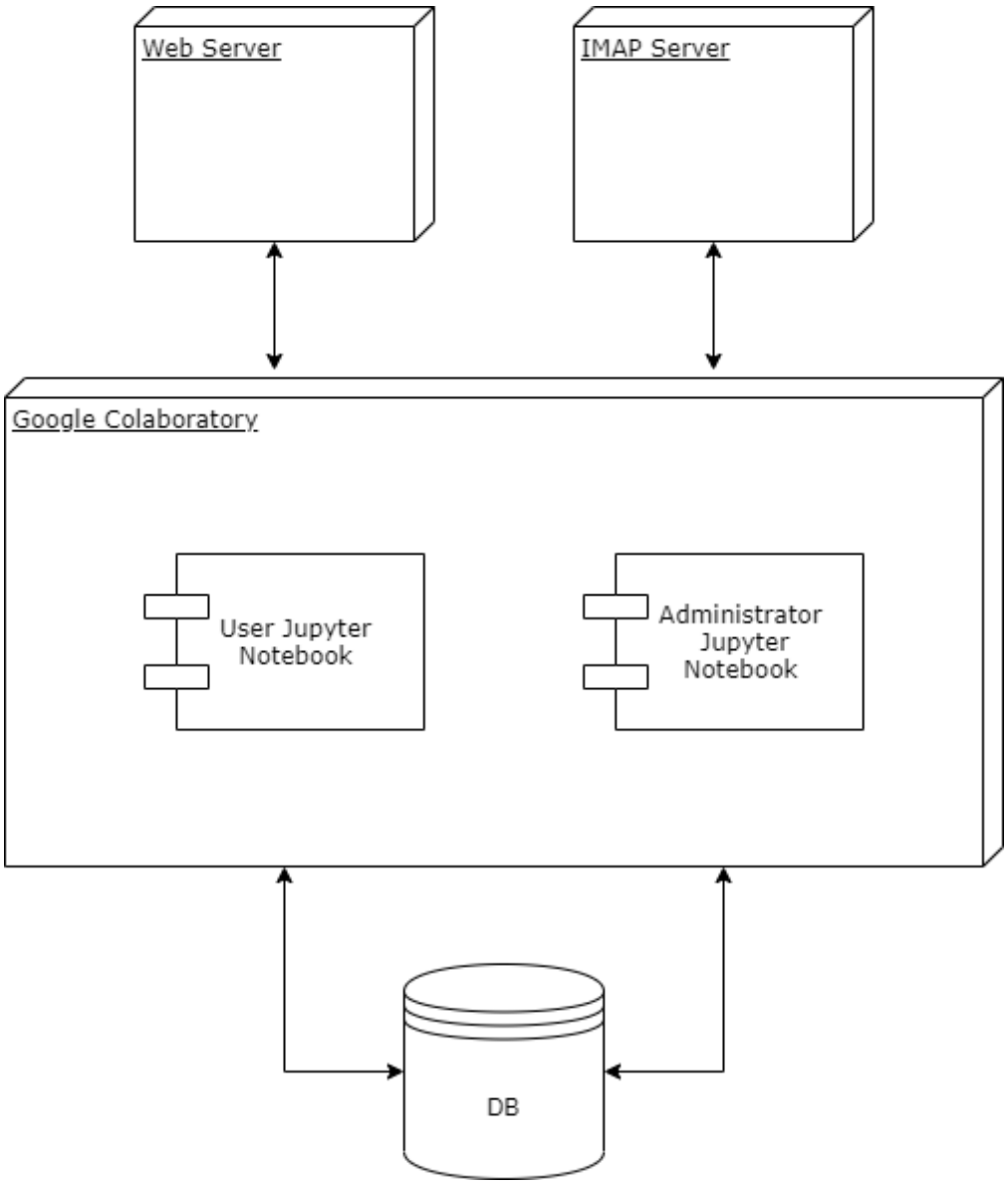


Figure 32. Physical View

6.2.5. SCENARIOS

The description of an architecture is illustrated using a small set of use cases, or scenarios, which become a fifth view. The scenarios describe sequences of interactions between objects and between processes. (Kruchten, 1995). The use cases descriptions and diagrams are shown in the previous chapter.

7. IMPLEMENTATION

7.1. DEVELOPMENT PLANNING

From the project design, we can identify a clear flow of information from the selection of Microsoft Office documents for analysis, through the extraction of macros embedded in the documents, as well as their obfuscation, to the extraction of features and classification using Machine Learning techniques.

We will follow this order in the development process, being able to see the evolution of the document analysis process. Thus, the first module to be developed will be the import and selection of documents module.

7.2. DEVELOPMENT ENVIRONMENT

We will develop the entire project in the Google Collaboratory service, its technical specifications are detailed below: (Google , 2020)

- GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM
- CPU: 1xsingle core hyper threaded Xeon Processors @2.3Ghz i.e(1 core, 2 threads)
- RAM: 12.6 GB
- Disk: 33 GB
- Lifetime: 12 hours

7.3. FILE IMPORT

This module has four subdivisions because the documents may come from four different sources. The benefit of using Google Colaboratory is the compatibility of this environment with the services of the same provider. The environment provides the user with an interface to import documents from the local device and an interface to connect with Google Drive accounts. To facilitate the use of these interfaces, we generated a user guide integrated into the Jupyter Notebook with the necessary actions for its use, recommending the use of Google Drive for the user's local device safety.

When downloading documents, the user is asked to enter the resource's URL. Once the URL has been entered, the program proceeds to download the resource, finally showing the resolution of the download and adding the document to the session storage.

Figure 33 Import form Internet

Importing documents from Gmail asks for the user's Gmail address and associated password, then asks for the source folder, the number of documents to be imported, and the extension of these documents. Finally, it stores the documents that meet these requirements in the session.

This development addresses several problems encountered during the implementation.

Firstly, by automating the user login process, Google views this activity as suspicious and sometimes blocks access to the account.

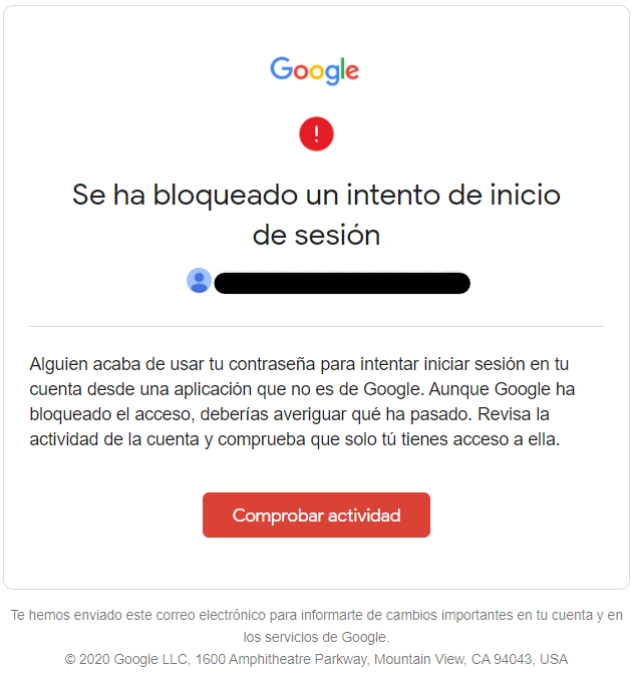


Figure 34. Import from Gmail

In second place, the import of documents implies allowing access to third party applications that generates a security breach in the Google account when the user enables this option. Moreover, this action is incompatible with additional authentication factors, making it an inconsistent action for the purpose of improving security.

In response to this problem, a guide is provided for the user to save the desired documents in the Google Drive storage unit and import them from this source. This is an alternative compatible with good user practices such as double authentication factors and maintains the isolation of potentially malicious documents from local systems without requiring access to third-party applications.

Additionally, a Python script will be provided with the requested functionality, which will allow the user to download attached documents to his local device, leaving a record that their use puts security at risk and may cause damage to the system.

7.4. FILE SELECTION

Once the user has imported the desired documents, the program must allow the user to select the documents to be analyzed. This functionality supports both document paths and folders containing Microsoft Office documents.

The implementation consists of requesting the paths and using a recursive function to extract the contained documents in case of a folder.This check is performed using the methods: ".isdir()" for folder checking and ".isfile()" for document checking. These methods belong to the "os" library, which we must include in the code dependencies.

When using a recursive function, the program performs an in-depth search on the specified folders, being able to extract documents that are in folders contained in the path.

The program finally displays the list of selected documents.

7.5. MACRO EXTRACTION

Macro extraction is the first step in document processing.

This functionality analyzes the structure of Microsoft Office documents looking for modules embedded in the documents, in order to extract the source code of the macros contained.

Microsoft Office documents have two formats for storing embedded objects:

- **OLE1.0:** “This format predates the OLE Compound File technology (as specified in [MS-CFB]). When using the OLE1.0 Format, the linked

object and embedded object data is laid out as a sequence of bytes within the container document." (Microsoft, 2020)

- **OLE2.0:** "This format uses the OLE Compound File technology (as specified in [MS-CFB]). When using the OLE2.0 Format, the container application creates an OLE Compound File Storage object ([MS-CFB] section 1.3) for each linked object or embedded object. The linked object or embedded object data is contained in this storage in the form of OLE Compound File Stream objects" (Microsoft, 2020)

OLE format specifies data structures in streams.

We will use the tool oledump.py (Didier Stevens Labs, 2020) in the version oledump_V0_0_43 compatible with Python3 for the extraction of the document streams and the macro source code extraction.

Attached is a result of stream extraction from a Word document obtained using the tool.

```
"/content/drive/My Drive/TFG/Emotet_files/attachments_8212305774.doc"
 1:      4096 '\x05DocumentSummaryInformation'
 2:       428 '\x05SummaryInformation'
 3:      6952 '1Table'
 4:    173281 'Data'
 5:        97 'Macros/Lrqysirhcnftl/\x01CompObj'
 6:       297 'Macros/Lrqysirhcnftl/\x03VBFrame'
 7:      8250 'Macros/Lrqysirhcnftl/f'
 8:       112 'Macros/Lrqysirhcnftl/i12/\x01CompObj'
 9:        44 'Macros/Lrqysirhcnftl/i12/f'
10:         0 'Macros/Lrqysirhcnftl/i12/o'
11:       992 'Macros/Lrqysirhcnftl/o'
12:       568 'Macros/PROJECT'
13: M   18936 'Macros/VBA/Kjbfmrjakm'
14: m    1172 'Macros/VBA/Lrqysirhcnftl'
15: M    1281 'Macros/VBA/Xvftphkfmnc'
16:     19319 'Macros/VBA/_VBA_PROJECT'
17:      1592 'Macros/VBA/__SRP_0'
18:       110 'Macros/VBA/__SRP_1'
19:       304 'Macros/VBA/__SRP_2'
20:       103 'Macros/VBA/__SRP_3'
21:       901 'Macros/VBA/dir'
22:      4096 'WordDocument'
```

Figure 35. Macro Extraction

Yellow "M" letter is the macro module indicator, the result shows modules 13 and 15 as macro containers.

We redirect the tool's output to a text document to identify the container modules. Once identified, we execute the extraction of the modules' source code and redirect their output by storing each VBA macro in a separate text document.

7.6. MACRO OBFUSCATOR

We know that one of the techniques used by Emotet to avoid detection is obfuscating the macros source code. We have discussed this technique in depth in the chapter dedicated to Emotet. However, obfuscation is a technique of code masking, which modifies the appearance while maintaining its functionality. It is therefore a technique that cannot be considered intrinsically malicious, as it can be used to hide non-malicious code.

Bearing this in mind, we can create a dataset with the macros extracted from malicious documents and another with the macros from benign documents, in order to obtain a measure of similarity, which will serve to discriminate the documents' nature using automatic learning models. The result will be a learning process focused on the classification of obfuscated and non-obfuscated documents, with our objective being to classify documents as either malicious or benign.

The solution for this problem is to calculate the similarity, comparing macros under the same conditions. This is why it is necessary to obfuscate macros that do not have an obfuscation layer.

Determining the obfuscation of a macro is a complex task, since the obfuscation process leaves no evidence.One of the possible solutions to detect whether a macro is cloudy is to use a method similar to the one explained above using similarity calculation. The problem with this solution is the time penalty associated with this calculation.

By comparing 50 macros for each dataset, the estimated time penalty amounts to one and a half minutes according to the tests performed [Reference test number in collaborative].

Another alternative is the study of the statement "Dim". This instruction is used for the variables assignment in the macros. It allows the name change of a variable and therefore has a frequent use when obfuscating macros.

This table shows the average of the "Dim" instructions detected in the macros of the malicious documents compared to the average of the benign documents, obtained from the analysis of the documents that make up the dataset.

Table 7-1. Dim instructions average

Dataset	Average
Malicious Documents	8.394017094
Bening Documents	2.102586207

Using this alternative, we only have to count the "Dim" instructions performed by the macro and if it is close to the average of the instructions in the malicious data set, it would not perform the obfuscation process.

Finally, the macros will be obfuscated using the VBA Obfuscator tool (Bonnet, y otros, 2018)

An example of macro obfuscation is shown below

Macro without obfuscation:

Macro obfuscated:

7.7. MACRO ANALYSIS

This section explains the extraction of features selected as discriminating characteristics.

It is important to emphasize that this process goes through the macro to be analyzed only once to reduce the execution time as much as possible.

Therefore, to show the complete implementation of this process, the operations are listed in order of execution.

For each macro in the document:

1. **line_count:** increases a counter for each line of the macro.
2. **operation_count:** increments one counter for each operation sum (+) found in the macro.
3. **max_concat_count:** calculates the maximum number of consecutive sum operations in a macro.
4. **string_count:** calculates the number of strings in the macro.
5. **string_size:** calculates the average length of the strings in the macro.
6. **asign_count:** calculates the number of assignments "=" of the macro.
7. **dim_count:** counts the number of "Dim" statements.
8. **for_count:** calculates the number of "For" loops in the macro.
9. **euclidean_distance_bad:** calculates the Euclidean distance of the macro's content with 40 malicious macros selected randomly from the dataset.
10. **euclidean_distance_good:** calculates the Euclidean distance of the content of the macro with 40 benign macros randomly selected from the dataset.
11. **tf-idf_distance_bad:** calculates the tf-idf metric of the content of the macro with 40 malicious macros selected randomly from the dataset.
12. **tf-idf_distance_good:** calculates the tf-idf metric of the content of the macro with 40 benign macros selected randomly from the dataset.

We then calculate for each document the average of values obtained in the calculation of its macros and we build a dictionary with the values to include them in the database.

7.8. TESTS

The following table lists the equivalency classes used to conduct the tests cases.

Table 7-2. Equivalency Classes for Tests

Requirements	Test ID	Description
RF_01: Import documents from local device	RF-01-E-CEV1	The file exists and it is in the specified path.
RF_03: Import documents from Internet	RF-03-E-CEV2	The specified URL exists and corresponds to a document
	RF-03-E-CEI1	The URL has an invalid format (https/http://xxx.xxx)
	RF-03-E-CEI2	The URL does not correspond to a web page.
	RF-03-E-CEI3	The URL exists but does not correspond to a document.
RF_04: Link the Google Drive account with the program	RF-04-E-CEV3	The authentication code is correct
	RF-04-E-CEI4	The authentication code is wrong
RF_05: Select documents for malware analysis	RF-05-E-CEV4	The selected path corresponds to an existing document.
	RF-05-E-CEV5	The selected path corresponds to an existing folder with documents.
	RF-05-E-CEI5	The selected path corresponds to an existing empty folder.
RF_06: Analyze selected documents	RF-06-E-CEV5	The selected document is valid for the analysis
	RF-06-E-CEI6	The selected document does not have macros
	RF-06-E-CEI7	The selected document does not have an Ole Format
	RF-06-O-CEV5	The selected document has obfuscated macros
	RF-06-O-CEV6	The selected document does not have obfuscated macros and obfuscate it

7.9. MODELS TRAINING

The training process of a machine learning model consists of establishing the appropriate values to accomplish the task for which it is intended. Our goal is to train our models to classify, using the extracted features, into malicious and benign documents. These two classes are defined below:

- Malicious: documents suspected of presenting evidence of Emotet's presence.

- Benign; documents that do not present suspicion of malicious behavior.

The training of the selected models involves a series of concrete actions that can be grouped as follows:

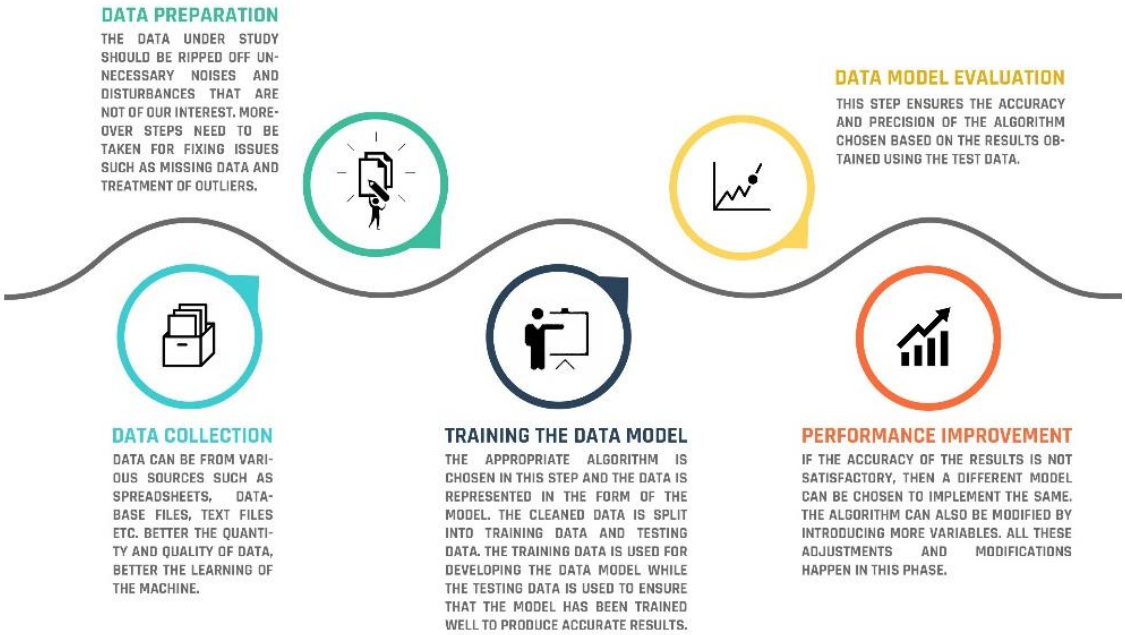


Figure 36 Model Training Process (Nath, 2016)

7.9.1. DATA COLLECTION

The data used for training the models comes from the dataset formed in the feature extraction process. This dataset is stored in the "features_extraction" collection of our MongoDB Atlas instance.

This collection is imported creating a Dataframe and using the function to_csv() from the pandas library, which generates a csv document with the characteristics selected for the training.

As it is a supervised training, it is necessary to specify the columns corresponding to the predicting variables and the variable to be predicted.

7.9.2. DATA PREPARATION

During this process it is usually necessary to check the data to remove empty tuples or unwanted values. In contrast, for reasons of database space optimization, the feature extraction process does store incompatible or empty documents and secures the values of all extracted features.

The next task of data pre-processing is to perform normalization. Normalizing means, in this case, compressing or extending the variable values so that they are in a defined range. To normalize our data we will use the MinMax Scaler method, normalizing the data in a decimal range between 0 and 1:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Figure 37. Normalization

Once normalized, all variables have the same range of values and generate a right learning between prediction variables.

7.9.3. TRAINING THE DATA MODEL

The next step is to divide our dataset into training set and test set.Remember that the training set is used to fit and tune the model and Test sets are used to evaluate the trained model.

This process determines to a great extent the model outcomes, a large number of patterns in the training set allows a precise adjustment of the model. However, the reduction of the test set can cause problems in the evaluation of this training.

Therefore, we will use cross validation of 10 sets to eliminate this conflict and additionally combat the reduced number of samples we have for learning.

The cross-validation method consists of dividing the initial dataset into 10 equal parts while maintaining the proportion of samples from the two classes to be classified. The training process is carried out in 10 iterations by selecting one of the 10 sets as a test set in each of these iterations. The resulting values are obtained from the average of the 10 iterations.

Figure 38. Cross- validation method (Norena, 2018)

The model is then trained.

7.9.4. DATA MODEL RESULTS

The model evaluation process consists of interpreting the values obtained from the training. These values show the acquired capacities from the training.

Depending on the purpose of the model, there are different ways to show the evolution. Our models are designed for binary classification, so we will use the following measures for this purpose:

Binary cross-entropy as loss function because it penalizes wrong classifications more than other functions such as MSE. This method maximizes the probability of correctly classifying documents.[7]

We will also consider the precision metric to observe the best of the model.

In order to obtain the best approach, four different MLP architectures will be analyzed and compared.

The tests made to choose the architecture model are on annexed tables at the end of the document.

[7] This article exceeds the scope of this paper but has been used for the selection of the metrics. It explains in a clear way the binary cross-entropy (Godoy, 2018)

Model 1:

This architecture is composed by two layers of 100 and 70 hidden neurons respectively, with a ReLU activation fucntion and an output layer with a sigmoid activation function.

Model 2:

This architecture is composed by three layers of 90, 60 and 60 hidden neurons respectively, with a ReLU activation fucntion and an output layer with a sigmoid activation function.

Model 3:

This architecture is composed by four layers of 50, 100, 100 and 100 hidden neurons respectively, with a ReLU activation fucntion and an output layer with a sigmoid activation function and dropout.

Dropout is an optimization strategy that ignores a series of neurons according to a probabilistic function, which "forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons." (Budhiraja, 2016)

Model 4:

This architecture is composed by three layers of 90, 60 and 60 hidden neurons respectively, with a ReLU activation fucntion and an output layer with a sigmoid activation function applying dropout strategy.

Model 5: SVM

We chose the model 2 with three layers without dropout for presenting the best results.

7.9.5. RESULTS ANALYSIS

Tests will then be made to decide which characteristics are selected for classification

Test 1:

In this first test, the features shown on the table below have been used to predict the documents nature.

Table 7-3 Test 1 Features

Features	Operation count	Max concat	String num	String size	Assign	For count	Euclidean	Tfidf distance	Euclidean distanc	Tfidf distance/line count

		count			count		distance		e/ line count	
	X	X	X	X	X	X			X	X

On the table below, the calculated values by each model are shown as a summary.

True positive and true negative reference to the hits the model has made.

False positive and false negatives are errors made by the model on the prediction. The false negative number is a very dangerous value because it means that an infected document would have been reported as benign. A high value on this field will be penalized during the election of the best model.

Table 7-4 Test 1 Results Summary

Loss	0.3719
Accuracy	0.8400
True Positives	122
False positive	12
True negative	136
False negative	37
Real positive	159
Real negative	148

TEST 2:

Table 7-5 Test 2 Model

Features	Operation count	Max concat count	String num	String size	Assign count	For count	Euclidean distance	Tfidf distance	Euclidean distance/ line count	Tfidf distance/line count
	X	X	X	X					X	X

Table 7-6 Test 2 Values

Loss	0.3831
Accuracy	0.8338
True Positives	132
False positive	20
True negative	124
False negative	31
Real positive	159
Real negative	148

TEST 3:

Table 7-7 Test 3 Model

Features	Operation count	Max concat count	String num	String size	Assign count	For count	Euclidean distance	Tfidf distance	Euclidean distance/ line count	Tfidf distance/line count
	X	X	X	X			X	X		

Table 7-8 Test 3 Values

Loss	0.3206
Accuracy	0.8100
True Positives	144
False positive	43
True negative	105
False negative	37
Real positive	159
Real negative	148

TEST 4:

Table 7-9Test 4 Model

Features	Operation count	Max concat count	String num	String size	Assign count	For count	Euclidean distance	Tfidf distance	Euclidean distance/ line count	Tfidf distance/line count
	X	X	X	X	X	X	X	X		

Table 7-10 Test 4 Values

Loss	0.4625
Accuracy	0.7800
True Positives	144
False positive	43
True negative	105
False negative	37
Real positive	159
Real negative	148

The final model selected is model two with the characteristics used in the first test

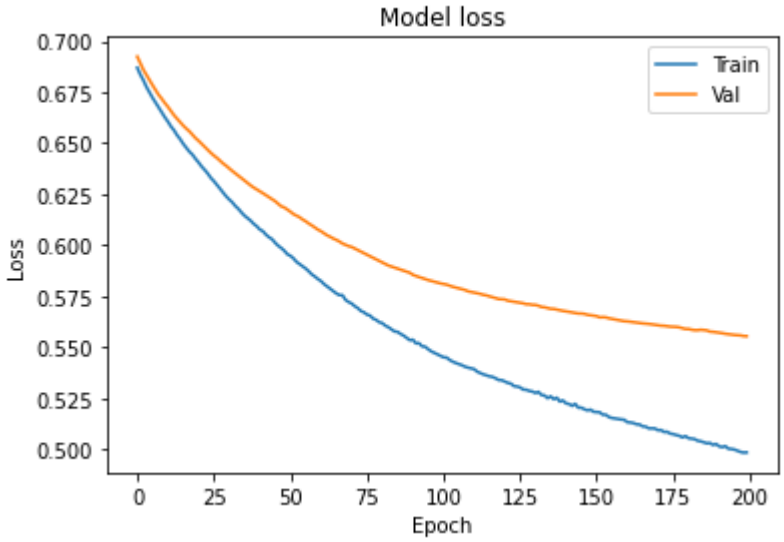


Figure 39. Final Model 2 plot

In addition the graphic presents a perfect fit structure.

8. CONCLUSION AND FUTURE LINES

8.1. CONCLUSIONS

From the socio-economic analysis of the problem, we can gather that the proposal of an effective detection method can have an incalculable impact for a great number of interested agents such as companies, governments, etc.

Analyzing in depth the nature of this threat we can also conclude that its detection only from the static analysis presents great limitations and difficulties due to the obfuscation techniques and the polymorphic nature of Emotet.

Despite these difficulties, the results obtained from the application of diverse solutions based on automatic learning present a promising perspective on the use of these techniques.

Based on the detection ratios obtained from the analysis of the results of our tool, we conclude that based on the static analysis of Microsoft Office documents, the presence of Emotet can be detected with 84% accuracy with 3% detection of false positives.

8.2. FUTURE LINES

A possible improvement of this tool could be using a dynamic model instead of a static one. This kind of model is better to detect the virus. However, it needs a better understanding as a base, and it is more expensive to create and maintain. For these reasons it has been thought as a possible future line of work.

Another future line could be the execution time improvement by using tensors to calculate the code similarity, being able to use graphic processors (GPUs) to speed up this task.

BIBLIOGRAPHY

Architectural Blueprints—The "4+1" View. **Kruchten, Philippe. 1995.** 1995, IEEE Software 12, pp. 42-50.

Bonnet, Nicolas and Leroy, Thomas. 2018. VBA Obfuscator. [Online] 2018. https://github.com/bonnetn/vba-obfuscator.

Bromium. 2019. *EMOTET: A TECHNICAL ANALYSIS OF THE DESTRUCTIVE, POLYMORPHIC MALWARE.* s.l. : Bromium INC., 2019.

Budhiraja, Amar. 2016. Medium. *Dropout in (Deep) Machine learning.* [Online] December 15, 2016. https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5.

Centro Criptológico Nacional. 2019. *Medidas de Actuación frente al código dañino EMOTET.* 2019.

Century Link. 2019. *Emotet Illuminated: Mapping A Tiered Botnet Using Global Network Forensics.* [Online] June 17, 2019. https://blog.centurylink.com/emotet-illuminated-mapping-a-tiered-botnet-using-global-network-forensics/.

Cybersecurity and Infraesctructure Security Agency. 2018. CISA. [Online] 07 20, 2018. https://us-cert.cisa.gov/ncas/alerts/TA18-201A.

Cylance. 2019. *2019 Threat Report.* 2019.

Diario Oficial de la Unión Europea. 2016. *DIRECTIVA (UE) 2016/1148 DEL PARLAMENTO EUROPEO Y DEL CONSEJO.* July 19, 2016.

Didier Stevens Labs. 2020. Didier Stevens Labs. [Online] 2020. [Cited: July 06, 2020.] https://blog.didierstevens.com/programs/oledump-py/.

ESET. 2019. *MACHINELEARNING ERA IN.* 2019.

Godoy, Daniel. 2018. Towards data Science. *Medium.* [Online] November 21, 2018. [Cited: August 1, 2020.] https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a.

Google . 2020. Colaboratory Frequently Asked Questions. [Online] 2020. https://research.google.com/colaboratory/faq.html.

Google Developers. 2020. Protocol Buffers. [Online] August 21, 2020. https://developers.google.com/protocol-buffers/docs/overview..

INTERPOL. 2020. *COVID-19 Cybercrime Analysis Report.* 2020.

JPCERT. 2019. EmoCheck la nueva utilidad para comprobar si un equipo está infectado con el malware tipo troyano Emotet. [Online] December 2019. https://www.cert.gov.py/noticias/emocheck-la-nueva-utilidad-para-comprobar-si-tu-equipo-esta-infectado-con-el-malware-tipo-troyano-emotet.

Kaspersky. 2019. *Kaspersky Security Bulletin 2019. Statistics.* 2019.

Kovacs, Kristof. 2020. kkovacs.eu. [Online] 2020.

2017. LegalToday. *Ciberataques : ¿cómo se regulan los nuevos peligros del siglo XXI?* [Online] September 8, 2017. https://www.legaltoday.com/practica-juridica/derecho-penal/penal/ciberataques-como-se-regulan-los-nuevos-peligros-del-siglo-xxi-2017-09-08/.

Lu, Kai. 2019. *A Deep Dive into the Emotet Malware.* June 6, 2019.

Mazzei, Patricia. 2019. Another Hacked Florida City Pays a Ransom, This Time for $460,000. *New York Times.* June 27, 2019.

Microsoft Defender ATP Research Team. 2018. *How artificial intelligence stopped an Emotet outbreak.* 2018.

Microsoft. 2020. Open Specifications. *OLE1.0 and OLE2.0 Formats.* [Online] 2020. https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-oleds/fdc5e702-d09e-4344-a77f-eb079d41f23f.

Nath, Deepu S. 2016. 9 Baby Steps to start with Machine Learning. *Medium.* [Online] August 6, 2016. [Cited: July 7, 2020.] https://medium.com/@deepusnath/9-baby-steps-to-start-with-machine-learning-fe3f31b83fe.

Norena, Sebastian. 2018. Towards Data Science. *Python Model Tuning Methods Using Cross Validation and Grid Search.* [Online] May 18, 2018. https://medium.com/@sebastiannorena/some-model-tuning-methods-bfef3e6544f0.

Sheehan, Daniel Patrick, Nerl, Daryl and Opilo, Emily. 2018. City of Allentown Computer systems hit by virus that will require nearly $1M fix. *The Morning Call.* February 20, 2018.

Signaturit. 2017. *What laws regulate cybersecurity in the European Union and in Spain?* April 26, 2017.

Sihwail, Rami, Omar, Khairuddin and Zainol Ariffin, Khairul Akram. 2018. *A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis.* 2018.

Sophos. 2019. Sophos News. *Emotet 101, stage 4: command and control.* [Online] March 5, 2019. https://news.sophos.com/en-us/2019/03/05/emotet-101-stage-4-command-and-control/.

Verizon. 2019. *2019 Data Breach.* 2019.

Wikipedia. 2020. WIkipedia the free Encyclopedia. [Online] June 27, 2020. https://en.wikipedia.org/wiki/Comparison_of_platform_virtualization_software.

ANNEXES

A. FUNCTIONAL REQUIREMENTS SPECIFICATION

Table 8-1 Functional Requirements Specification

ID	NAME	DESCRIPTION	PRIORITY	PRE-CONDITION	POST-COSDITIONS	NF REQS	NOTES
RF_01	Import documents from local device	The program allows the agent to select Microsoft Office documents stored on his local device and import them into the program	M	- Have internet access - Have the documents for analysis on the local device	Imported documents will be displayed in the program		
RF_02	Import documents from Gmail	The program allows the agent to select Microsoft Office documents from his local device and perform the malware analysis, obtaining a report of Emotet's malware detection	M	- Have internet access - Have a Google Account - Have enabled the Imap option on the Google account - Have enabled the Third-party apps with account access option on the Google account	Imported documents will be displayed in the program		
RF_03	Import documents from Internet	The program allows the agent to specify the URL of a Microsoft Office document and download the file from the internet	D	Have internet access	Downloaded documents will be displayed in the program		

RF_04	Link the Google Drive account with the program	The program allows the agent to link his Google Drive storage unit using Google Account authentication method	M	- Have internet access - Have a Google Account - Have enabled the Third-party apps with account access option on the Google account	Google Drive unit will be visible in the program
RF_05	Select documents for malware analysis	The program allows the agent to select the path of Microsoft Office documents or folder that contains Microsoft Office documents to execute the malware analysis over the selection	M	Have the documents selected on the program	Selected paths or folders will be displayed in the program
RF_06	Analyze selected documents	The program allows the agent to analyze selected Microsoft Office documents	M	- Have the documents for analysis on the program - Have MLP Model Trained - Have SVM Model Trained	Analysis report will be displayed in the program
RF_07	Download malware Analysis report	The program allows the agent to download the malware analysis report on his local device	D	Have the documents for analysis on the program Have MLP Model Trained Have SVM Model Trained	Analysis report will be displayed in the program

B. NON- FUNCTIONAL REQUIREMENTS SPECIFICATION

ID	NAME	DESCRIPTION	NOTES
RNF_01	Isolated Enviroment	Program execution must be isolated from adjacent systems to prevent malware infections	
RNF_02	Execution_time	The program execution time should be time as optimized as possible.	
RNF_03	Report terminology	The report should be readable by any user with a simple language and reduced technical terminology.	

C. REQUIREMENTS TRACEABILITY MATRIX

Table 8-2. Requirements Traceability Matrix

FUNCTIONAL REQUIREMENTS	USE CASES					
	UC_01 Import from device	UC_02 Import from Gmail	UC_03 Import Google Drive	UC_04 Import from Internet	UC_05 Analyze document	UC_06 Dowload report
RF_01 IMPORT FROM LOCAL DEVICE	X					
RF_02 IMPORT FROM GMAIL		X				
RF_03 IMPORT FROM INTERNET				X		
RF_04 LINK WITH GOOGLE DRIVE			X			
RF_05 SELECT DOCUMENT					X	
RF_06 ANALYZE DOCUMENT					X	
RF_07 DOWNLOAD REPORT						X

D. PLANNING GANNT DIAGRAM

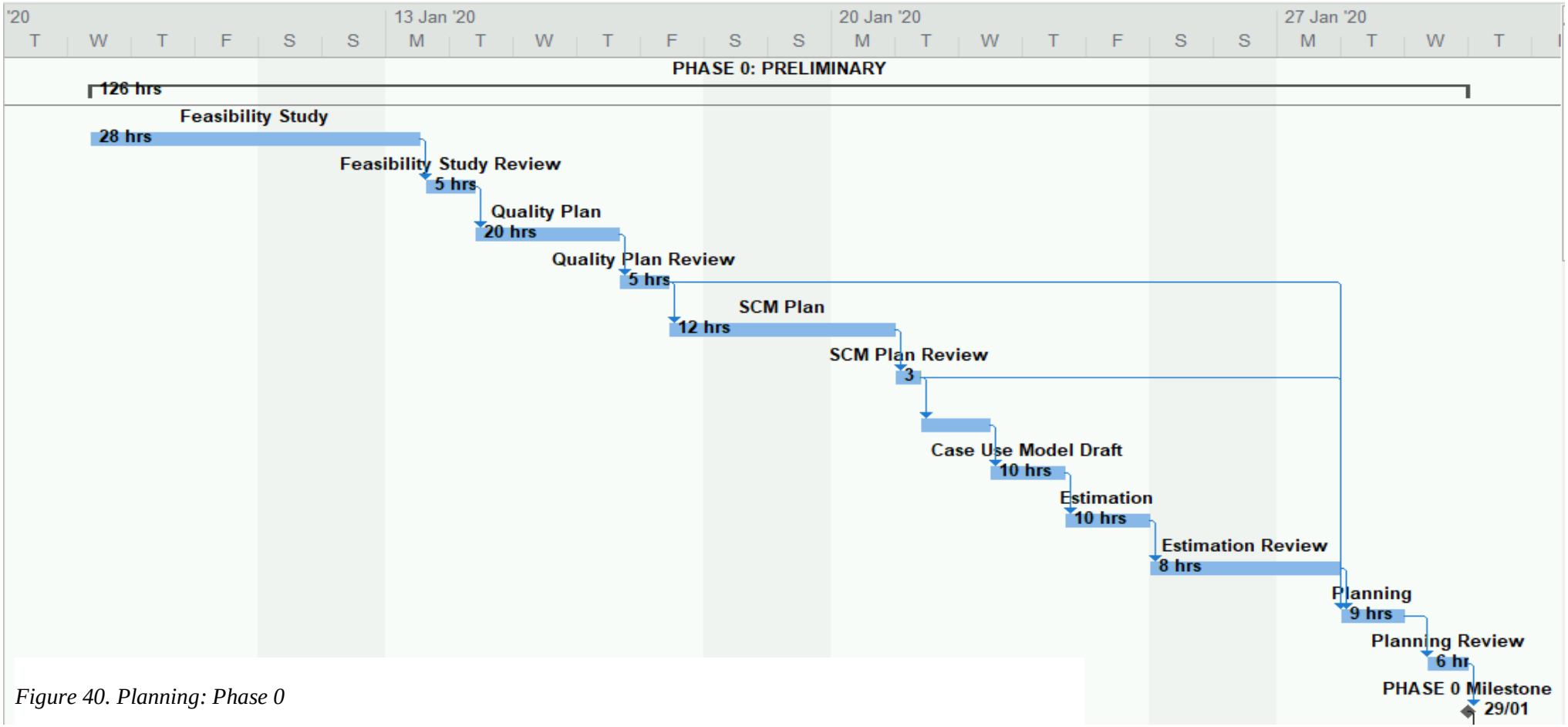


Figure 40. Planning: Phase 0

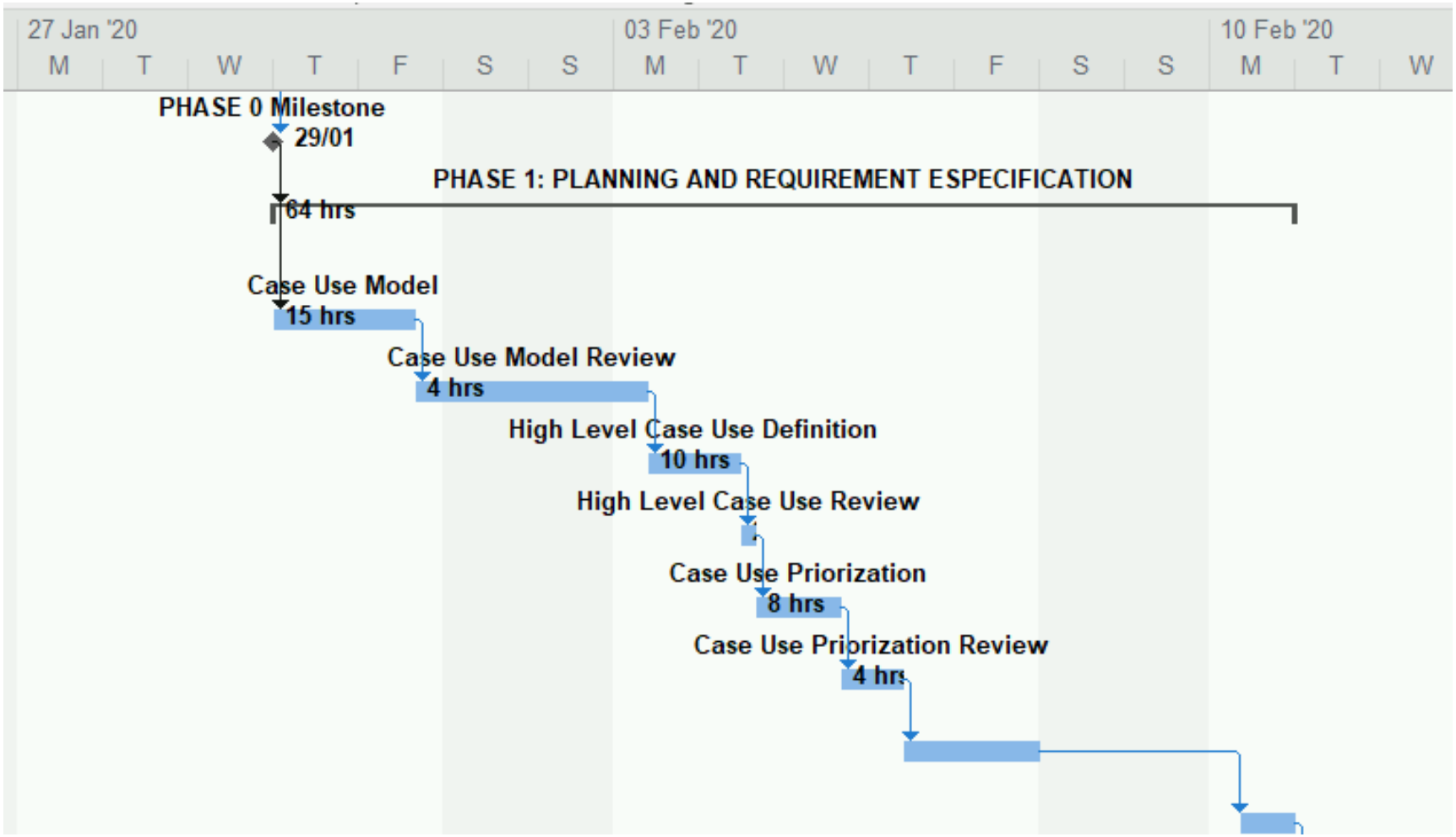


Figure 41. Planning: Phase 1

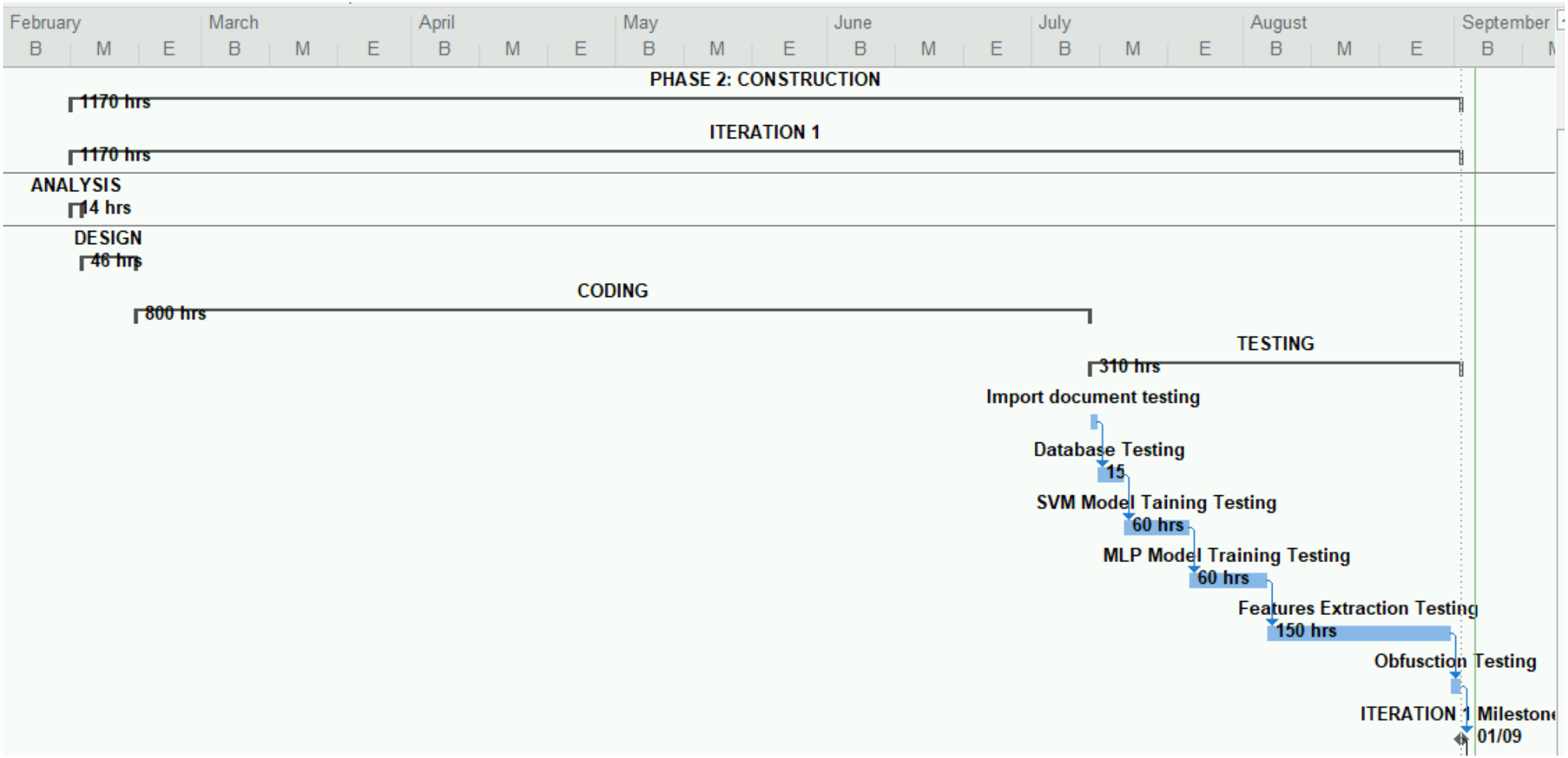


Figure 42. Planning: Phase 2

A. Model 1: Architecture choosing

Table 8-3 Model Architecture choosing

Architecture test	Layer 1 number of neurons	Layer 2 number of neurons	Loss	Accuracy
1	100	70	0.4356	0.8200
2	100	50	0.4439	0.8000
3	100	30	0.4512	0.8200
4	100	10	0.4594	0.8200
5	100	80	0.4392	0.8000
6	100	100	0.4469	0.8000
7	100	120	0.4498	0.8200
8	10	70	0.4610	0.8200
9	30	70	0.4712	0.8200
10	50	70	0.4646	0.8200
11	70	70	0.4498	0.8000
12	120	70	0.4442	0.8200
13	140	70	0.4514	0.8000
14	70	100	0.4535	0.8200

B. Model 2:

Architecture test	Layer 1 number of neurons	Layer 2 number of neurons	Layer 3 number of neurons	Loss	Accuracy
1	80	60	60	0.3493	0.8600
2	80	60	20	0.3401	0.8000
3	80	60	80	0.3818	0.8800
4	80	60	100	0.3308	0.8200
5	80	30	60	0.3600	0.7800
6	80	80	60	0.3514	0.8000
7	80	100	60	0.3531	0.8000
8	100	60	60	0.3667	0.8800
9	90	60	60	0.3100	0.8800
10	80	60	60	0.3590	0.7800
11	40	60	60	0.3867	0.8000
12	120	80	80	0.3654	0.8200
13	60	50	50	0.3500	0.7800

Table 8-4 Model 2: Architecture choosing

C. Model 3: Architecture choosing 4 layers with dropout

Table 8-5 Model 3: Architecture choosing

Architecture test	Layer 1 number of neurons	Layer 2 number of neurons	Layer 3 number of neurons	Layer 4number of neurons	Loss	Accuracy
1	50	100	100	50	0.5812	0.8600
2	50	100	100	30	0.0.5950	0.8400
3	50	100	100	80	0.5699	0.8600
4	50	100	100	100	0.5672	0.8600
5	50	100	100	140	0.5621	0.8400
6	50	100	100	100	0.5470	0.8600
7	50	100	80	100	0.5596	0.8400
8	50	100	120	100	0.5782	0.8200
9	50	80	100	100	0.5768	0.8200
10	50	120	100	100	0.5677	0.8200
11	20	100	100	100	0.5570	0.8400
12	70	100	100	100	0.5604	0.8600
13	100	100	100	100	0.5630	0.8400

D. Model 4: 3 layers with dropout

Table 8-6 Model 4: Architecture choosing

Architecture test	Neuronas capa 1	Neuronas capa 2	Neuronas capa 3	Loss	Accuracy
1	90	60	60	0.5513	0.8600
2	90	60	100	0.5530	0.8600
3	90	60	30	0.5880	0.8200
4	90	90	60	0.5560	0.8200
5	90	40	60	0.5637	0.8600
6	120	60	60	0.5568	0.8200
7	70	60	60	0.5484	0.8400

E. Model 5: SVM

Table 8-7 Model 5:SVM

Loss	Accuracy
0.5513	0.8000
0.5530	0.8400
0.5880	0.8200
0.5560	0.8100
0.5637	0.8300
0.5568	0.8200
0.5484	0.8400