Eduardo
Fernandes

Integrar Inteligência Artificial para melhorar a capacidade de análise de e-mails fraudulentos

Streamlining the analysis of phishing emails using Artificial Intelligence

PROPOSTA DE TESE

Eduardo
Fernandes

Integrar Inteligência Artificial para melhorar a capacidade de análise de e-mails fraudulentos

Streamlining the analysis of phishing emails using Artificial Intelligence

PROPOSTA DE TESE

"*The greatest challenge to any thinker is stating the problem in a way that will allow a solution*"

— Bertrand Russell

Universidade de Aveiro
2024

Eduardo Fernandes

Integrar Inteligência Artificial para melhorar a capacidade de análise de e-mails fraudulentos

Streamlining the analysis of phishing emails using Artificial Intelligence

Proposta de Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à conclusão da unidade curricular Proposta de Tese, condição necessária para obtenção do grau de Mestre em Engenharia Informática , realizada sob a orientação científica do Doutor João Almeida, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Sérgio Matos, Professor professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Texto Apoio financeiro do POCTI no âmbito do III Quadro Comunitário de Apoio.

Texto Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

Dedico este trabalho aos meus pais e amigos pelo incansável apoio.

o júri / the jury

presidente / president

Prof. Doutor João Antunes da Silva
professor associado da Faculdade de Engenharia da Universidade do Porto

vogais / examiners committee

Prof. Doutor João Antunes da Silva
professor associado da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor João Antunes da Silva
professor associado da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor João Antunes da Silva
professor associado da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor João Antunes da Silva
professor associado da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor João Antunes da Silva
professor associado da Faculdade de Engenharia da Universidade do Porto

agradecimentos / acknowledgements

Agradeço toda a ajuda a todos os meus colegas e companheiros.

Palavras Chave

e-mail, deteção de phishing, processamento de linguagem natural, análise de sentimento, inteligência artificial.

Resumo

A constante evolução e frequência dos ataques de phishing por e-mail representam um desafio significativo para a cibersegurança. Esta tese explora a integração da Inteligência Artificial (AI), especificamente técnicas de Processamento de Linguagem Natural (NLP) e Aprendizado de Máquina (ML)/Aprendizado Profundo (DL), para melhorar a detecção e análise de e-mails de phishing. Utilizando módulos de NLP impulsionados por AI, este estudo tem como objetivo criar uma solução em AI que não apenas detecta com precisão e-mails de phishing, mas também automatiza as capacidades de resposta, melhorando assim a eficiência e eficácia das equipes de CSIRT. Em última análise, esta pesquisa contribui para o campo da cibersegurança, fornecendo um quadro abrangente e alimentado por AI para uma detecção e resposta a e-mails de phishing mais robusta e adaptável.

Keywords e-mail, phishing detection, natural language processing, sentiment analysis, artificial intelligence.

Abstract

The increasing sophistication and frequency of email phishing attacks pose a significant challenge to cybersecurity. This thesis explores the integration of Artificial Intelligence (AI), specifically Natural Language Processing (NLP) and Machine Learning (ML)/Deep Learning (DL) techniques, to enhance the detection and analysis of phishing emails. By using AI-driven NLP modules, this study aims to create an AI-based solution that not only accurately detects phishing emails but also automates response capabilities, thereby enhancing the efficiency and effectiveness of CSIRT teams. Ultimately, this research contributes to the cybersecurity field by providing a comprehensive, AI-powered framework for more robust and adaptive phishing email detection and response.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Excertos de Código

Glossário

CAPÍTULO 1

Introduction

In today's world, the internet is present in our daily lives. The internet has evolved from a research and communication tool to an essential element of almost everything. Instant access to information, global communication and entertainment has become an integral part of our daily routine. Currently, cyberspace serves as the primary space for various economic, commercial, cultural, social and governmental activities and interactions. This space is intertwined with various parts of our existence, and any instability, insecurity or challenges within it may directly impact different areas of human lives [2].

As the internet continues to grow, some instability and risks can follow this growth, which may be reflected in more issues, or subsequently potential cyber threats. Malicious actors want to take advantage of those vulnerabilities to conduct cyber attacks, aiming to access confidential information and harm individuals, institutions, or companies [3]. These malicious activities are not only disruptive but can also result in substantial financial losses and breaches of sensitive information. However, there are several forms of cyber attacks, from the spread of malware and ransomware to data invasion.

One of the potential strategies to take advantage of the systems can be using people who are part of the institution. To deceive those, attackers may adopt social engineering techniques to prompt individuals to make decisions without much thought about what is happening, which can be advantageous when they are exploiting vulnerabilities in those processes. One common form of social engineering, that still has a high impact on organizations, is phishing [4].

1.1 Definition of phishing

The phishing attacks attempt to access confidential information to harm people, institutions or companies. Phishing is a type of cyber attack that is the combination of social engineering and technology to gain access to restricted information of end users [5]. Phishers or attackers try to mislead people into giving away their private information by illegally using a public or trustworthy organization. By posing as legal organizations, attackers can lure victims into clicking some malicious link that provides sensitive information to the attacker. There are

several types of phishing attacks, but the most popular are those that use communication channels such as emails and SMS (smishing) to trick users. Email is one of the most common forms of electronic communication, both in formal and informal situations. Therefore, email services are frequent targets of phishing attacks. In these attacks, attackers create fake emails that look real but are trapped to trick the user into stealing information or carrying out other types of malicious attacks.

According to the latest 2022 report from the **apwg!** (**apwg!**), 2022 was a record year with around 4.7 million phishing attacks. This is an increase of 150% per year since 2019 [6]. In the **apwg!** report for the 3rd quarter of 2020, it is mentioned that the number of phishing attacks has grown since March 2020. One major influence in the increase of phishing attacks since then is the COVID-19 pandemic [7]. As the subject of the pandemic was very present in everyday life and with the global lockdown, meaning that a very large number of people were at home, the attackers used texts related to COVID-19 in their attacks to make more victims fall into the trap. According to the ENISA report on phishing, *"They either falsely claimed to showcase of infection in the victim's area or shared medical experts' opinions to lure the victim to follow a malicious link"* [8].

The phishing problem is a major threat to all kinds of users on the internet and could lead to financial losses. Nowadays there are a huge number of businesses that suffer from this type of cyber attack. As stated by ENISA, there were 26.2 billion dollars of losses in 2019 due to the **bec!** (**bec!**) attacks. In their report, they concluded that 86% of global organizations suffered **bec!** attacks, which demonstrates the gigantic problem that companies around the world are repeatedly exposed to [8]. However, it is not just the business sector that is exposed to these attacks. In 2019, the health sector, government and public administration entities were also severely affected by phishing attacks, with even Ukrainian diplomats falling victim to fraudulent emails [8].

1.2 Motivation

Nowadays, phishing attacks have become one of the most prevalent cybersecurity threats faced by institutions and individuals alike. As attackers develop increasingly sophisticated methods, it is difficult to distinguish between genuine and malicious communications. For larger institutions, this problem is worse. Every day, countless emails flow into the inboxes of its members, and while built-in filters manage to flag some of these as phishing attempts, personalized attacks often go unnoticed. This puts sensitive data at risk and can lead to significant financial and reputational damage if not resolved quickly.

Current methods for identifying and combating phishing attacks, especially at large-scale institutions, face limitations. Automated filters, based on predefined criteria, may fail if new phishing techniques are introduced. At the same time, human-driven interventions, such as the **csirt!** (**csirt!**), face challenges of scalability. As the volume of potential threats grows, manually analyzing and addressing each suspected email becomes time-intensive and can lead to delays in response, giving attackers a huge advantage.

The constant evolution of phishing attacks requires a dynamic solution that can adapt and respond in real-time. **ai!** (**ai!**), with its **nlp!** (**nlp!**) capabilities, offers a possible solution to this problem. By automating the process of email analysis, we can not only detect potential threats with increased accuracy but also ensure timely responses, thus minimizing potential damages. Additionally, integrating AI-based expertise with human expertise, like that of the **csirt!** members, can result in a robust and comprehensive approach to combating phishing.

1.3 Objectives

The rapid growth of phishing attacks, as well as the problems they cause, indicate the need for an innovative way of detection and response. By utilizing the power of **ai!**, this study seeks to explore, design, and test an innovative framework to streamline the analysis of phishing emails.

One of the objectives is to gain an understanding of the techniques and methods commonly used by cyber attackers in phishing attacks. This study aims to examine AI-driven **nlp!** modules and assess their relevance and potential, for analyzing phishing emails.

The primary goal is to create an AI-based solution that can accurately detect phishing emails by utilizing **nlp!** techniques. An integrated system that not only identifies phishing emails but also automates response capabilities improving the efficiency and effectiveness of CSIRT teams.

The applicability of the framework will be evaluated, in a use case, using phishing emails as test data from the Cybersecurity Office known as GCS.

By accomplishing these objectives we aim to answer the research question: *How can Artificial Intelligence be integrated to enhance the detection and analysis of phishing emails and improve the response capabilities of the CSIRT teams?*

1.4 Dissertation Outline

The remaining parts of this dissertation are structured into several key chapters, and are organized as follows: Chapter 2 presents the State-of-the-Art that analyze the current landscape in phishing email detection. In chapter 3, it is proposed the future work plan of this thesis.

CAPÍTULO 2

State-of-the-art

This dissertation aims to develop a tool capable of improving the ability to analyze fraudulent emails. Given this problem, it is necessary to investigate the main **ai!** tools currently used in this context. This involves a comprehensive understanding of their capabilities and functionalities. Techniques and strategies for analyzing phishing emails, with an emphasis on **ai!** and machine/deep learning algorithms, and email data processing using **nlp!** modules, are also examined. These topics will be discussed in the sections below.

To carry out this investigation it is necessary to have good sources of information. Several articles from scientific journals and conferences were researched, so it was necessary to create some criteria to condense all the important information. Articles with a recent date are one of the most important parameters to take into account when filtering them. The cybersecurity domain is dynamic, with attackers constantly developing new techniques and tactics. If more recent articles are prioritized, the search is guaranteed to reflect the current state of phishing attacks and the latest strategies to resolve the problem.

Another criterion was to restrict to cyber phishing attacks only. By focusing exclusively on phishing, we aim to ensure the methodologies and results presented are directly relevant to the specific challenges of phishing attacks.

For research to be valuable, it needs to demonstrate effectiveness in detecting phishing attempts, and prioritizing articles that demonstrate good results ensures that the methodologies presented are effective and can serve as a reference.

2.1 E-mail Feature Engineering

Millions of emails are sent daily, making email a popular form of contact for all people around the world. Today, having one or more email addresses is considered normal, with email becoming just as common as phone calls for communication [9]. However, the extensive use of email as a main form of communication also brings with it certain special risks. The very aspects that make email a versatile and essential medium like its ease of use, immediacy, and the ability to reach a wide audience quickly, also make it an attractive platform for

malicious actors. Phishing attacks, in particular, exploit the trust and routine nature of email interactions. Because they are used to receiving legitimate emails regularly, users might not always examine every message carefully, especially when it is expertly written to look like real correspondence. This issue is made worse by the massive volume of information that is sent via email, known as email overload [10], which raises the probability that deceptive emails will be ignored. As a result, the same qualities that have made email a mainstay of modern communication also make it an ideal environment for phishing attacks, calling for sophisticated detection systems to separate authentic communications from fake ones.

2.1.1 What is an email?

Email, short for electronic mail, is a method of exchanging digital messages across the Internet or other computer networks. It remains an essential platform for electronic communication and a necessary tool for social relationships. Originally intended as a tool for basic text communication, email has developed into an essential element of modern communication in both private and professional environments, being used within organizations to exchange information and coordinate action, as well as by ordinary people to talk with friends [11]. Emails can be used for several things, such as information exchange, sending greetings and invitations, sending links to websites, or sending digital files (such as simple Word documents, images, and videos). Its use and functionality have been standardized by some important protocols that define the mechanism of the email exchange between servers and clients, allowing them to travel across the network correctly. That being said, enabling both incoming and outgoing email messages involves three specific protocols: **smtp!** (**smtp!**), **pop3!** (**pop3!**), and **imap!** (**imap!**).

Defined in **rfc!** (**rfc!**) 5321 [12], **smtp!** is the standard protocol for email transmission across the Internet. It outlines how **mtas!** (**mtas!**) relays messages from the sender to the recipient's server. SMTP servers and clients provide a mail transport service and therefore act as **mtas!**. **pop3!** and **imap!** are protocols for receiving email messages and operate in different ways to retrieve or access to email messages. While the **imap!** protocol allows simultaneous access by multiple clients, **pop3!** assumes that your email is being accessed only from one application. When a **pop3!** client connects to the mail server, it retrieves all messages from the mailbox, keeping them on the local device and erasing them from the server. On the other hand, **imap!** keeps the messages on the server and synchronizes the local device with the server. This means that the messages are stored on the server and can be accessed from multiple devices.

Figure **??** shows an example of the email delivery process between the sender and the recipient. This flow is explained in the following steps:

1. The sender writes an email and clicks the send button;
2. The email's destination must be determined by the **smtp!** server. It makes a DNS query to find data related to the recipient's information;
3. The DNS server returns the necessary information of the recipient's email service provider to the **smtp!** server;
4. The **smtp!** server sends the email across the Internet to the destination mailbox;

5. In this stage, the email passes through various **smtp!** servers and is finally relayed to the destination **smtp!** server;
6. The email finally reaches the final **smtp!** server;
7. The sender email is forwarded and is now sitting in the local **imap!**/**pop3!** server waiting for the recipient;
8. Upon logging into his email client, the intended recipient checks for fresh emails in his mailbox by querying the local **imap!**/**pop3!** server;
9. The receiving email client copies (**imap!**) or downloads (**pop3!**) the sender email.

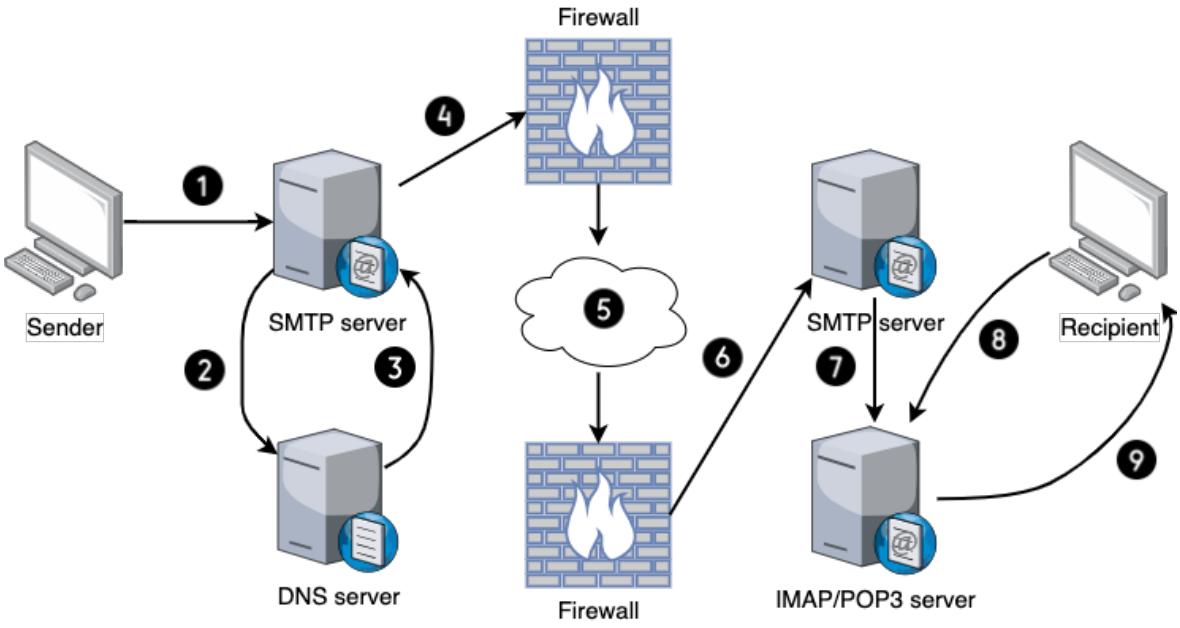


Figura 2.1: How email travels from the sender to the recipient.

2.1.2 Email structure

Email communication is an integral component of modern digital communication, and now we know how this communication happens, understanding how an email travels from point A to point B and the protocols involved in the process. However, it is also important to understand the structure of an email and the information it contains. The standard format of email messages is known as **imf!** (**imf!**). As specified in **rfc!** 5322 [13], it defines the required headers and bodies for messages, as well as the content and syntax for different headers. There is also the **mime!** (**mime!**) standard, which extends the capabilities of email to include multimedia content and non-ASCII text. It allows for the formatting of multipart messages and the inclusion of various types of binary files like images and documents.

Such protocols and formats led to the development of various email storage and exchange formats, notably **eml!** (**eml!**) and **mbox!** (**mbox!**). These formats utilize the foundational principles of these protocols to manage and store email data effectively.

The **eml!** format typically stores each email message as an individual file, incorporating the standardized headers and body prescribed by the **imf!**. Attachments in **eml!** files are either included as **mime!** content within the message or referenced as separate files. **mbox!**

combines all the emails in a folder into a single file. Although **eml!** and **mbox!** have gained widespread acceptance as standard formats because of their interoperability with current email clients, their approaches to email storage are different. Considering that **mbox!** is a method that keeps several emails in a file, handling each one separately may provide issues, while the individual file storage in **eml!** offers more granularity. Also, it is appropriate to address the **pst!** (**pst!**) format, which is primarily utilized by Microsoft Outlook. **pst!** files contain not just emails but also contacts, tasks, notes, and calendar events all in one file.

The selection of email format is crucial for efficient data administration and analysis when creating a system for phishing email detection. The decision to choose the **eml!** format over **mbox!** is motivated by the particular advantages it provides, especially about the granularity, providing large information about an email. The standardized nature of **eml!** files ensures broad compatibility with a variety of email clients beyond Microsoft Outlook, which is not the case with the **pst!** format. For a phishing email detection system that would need to process data from several sources, this compatibility is essential.

eml! files include all of the raw data that makes up an email, including the headers and the body content.

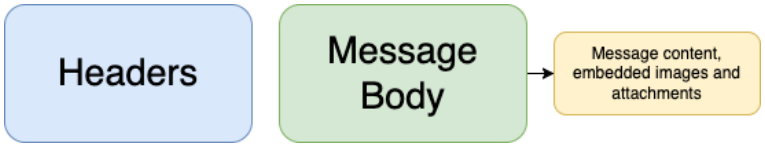


Figura 2.2: Contents in **eml!** file.

The email headers contain details about the email servers that carried the email, thus serving as a digital trail of the email's journey from sender to recipient. This header is not just a single entity but a collection of various fields, each holding specific information. Header fields are lines beginning with a field name, followed by a colon (":"), and followed by a field body, as specified in **rfc!** 5322 [13]. The field name identifies the type of information, and the field body, following the colon, contains the specific details corresponding to the field name. An example of an email headers message as an **eml!** file can be found in Figure **??**. Several fields are present in the email headers, each with its own purpose:

A **Delivered-To**: The intended recipient's email address is contained in this email header field;

B **Received By**: This field contains the details of the last visited **smtp!** server, where the information revealed is the Server's IP address, **smtp!** ID of the visited server, and data and time at which the email was received by the **smtp!** server;

C **X-Received**: This field shares the IP address of the message-receiving servers, the **smtp!** ID of the server, and the date and time at which the email was received;

D **Return Path**: The return path is an email header that tells **smtp!** servers where they should send non-delivery notifications. According to RFC 5321, [12], the return path consists of the sender's mailbox;

E **Received From**: It has some information about the IP address of the sender along with other details like the hostname. Every server that handles this mail adds this header;

F **Received-SPF**: The system forwards the message only after the sender's identity is authenticated with the **spf!** (**spf!**). **spf!** is designed to verify that the sending server is authorized to send emails on behalf of the domain in the "From"address. It uses the domain address for authentication and adds the check status in the header field;

G **Authentication Results**: **mtas!** apply a slew of authentication techniques to the email messages before processing them and add the results to this header field. It shares the ID of the authentication-performing server, the authentication techniques along their results;

H **From**: This field contains the sender's email address, indicating who sent the email;

I **To**: This field contains the recipient's email address;

J **Subject**: The subject line of the email offers a summary or a title to the email's content;

K **Date**: This indicates when the email was sent, providing a timestamp for the communication;

L **Message-ID**: It is the email's distinct ID that allows for differentiation. The same message ID cannot be shared by two emails;

M **MIME-Version**: This demonstrates that the message is prepared with the Multipurpose Internet Mail Extension (MIME) and supports a variety of forms, including audio, video, and plain text files.

Depending on the email delivery service, custom headers can be included and are called X-Headers. The primary purpose of X-headers is to address the specific requirements of the sender that are not covered by the standard headers.

Figura 2.3: Email headers as an **eml!** file example.

As was discussed previously, the email content offers a wide range of information that can be crucial for detecting phishing emails. Besides the headers, the email body also contains equally pivotal information for detecting phishing attempts. While the email headers provide critical metadata, the body of an email often contains the substantive content that is essential for a more comprehensive analysis.

Contents of the email body are described by its "Content-Type"field, which indicates the respective formats of the information. The structure of the "Content-Type"consists of a "type"and a "subtype", two strings, separated by a '/', where no space is allowed between them. The type represents the category and can be a discrete or a multipart type, and the subtype is specific to each type. Discrete types are types that represent a single file, such as a single text or music file, or a single video. A document that is divided into several separate sections, each of which could have its own unique MIME type, is represented by a multipart type.

The list of discrete types is long but some important content-types are mentioned below:

- **text**: Represents format which is human-readable. Includes subtypes such as "text/plain", "text/html", "text/css", and "text/javascript";
- **image**: Represents image of any type. Common subtypes examples are "image/jpeg", "image/png", and "image/svg+xml";

- **audio**: Represents any audio file format. Subtypes examples include "audio/mpeg", and "audio/wav";
- **application**: Represents any kind of binary data. Generic binary data is represented with the "application/octet-stream"subtype. Other common examples include "application/pdf", and "application/zip".

```
--_000_f7a1e5acac5e44248e649c504db817e6uapt_
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Dear member,


We have faced some problems with your account. Please update the account. I=
f you do not update will be Closed.


To update your account, just confirm your information.(It only takes a minu=
te)

It's easy:

  1.  Click the link below to open a secure browser window.
  2.  Confirm that you're the owner of the account, and then follow the ins=
tructions.

Relog in your account now<http://www.google.com>

--_000_f7a1e5acac5e44248e649c504db817e6uapt_
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

Figura 2.4: Email body in an **eml!** file example.

Expanding the in-depth analysis of the email body, it is equally important to delve into the structure and representation of attachments in **eml!** files. The attachments contained in the email file are encapsulated within the message body, marked by their "Content-Type"(**A**) and "Content-Disposition"(**B**). The **A** header specifies the media type of the file, which is crucial in understanding the nature of the attachment. The following example in Figure **??** indicates that the attachment is a PDF file, with a given name ("matter.pdf"). Also, the **B** header plays a critical role in how the attachment is processed by the email client, suggesting that the file should be treated as an attachment and not as an inline element. The 'filename' parameter provides the suggested name for the file when saved. Finally, the "Content-Transfer-Encoding"(**C**) header indicates that the file is encoded in base64, which is a common encoding technique that ensures the binary data of the attachment is transmitted over the network in a text format.

```
A Content-Type: application/pdf; name="matter.pdf"
  Content-Description: matter.pdf
B Content-Disposition: attachment; filename="matter.pdf"
C Content-Transfer-Encoding: base64
```

Figura 2.5: Attachment example in **eml!** file.

2.1.3 Email features for phishing detection

Email metadata plays a critical role in the field of email phishing detection. All the fields explained before, including headers and other structural components, can offer information to determine the authenticity of an email. This metadata, which users frequently ignore, includes details such as the sender's address, routing information, timestamps, and more, giving information to help comprehend an email's origin and path.

Email spoofing is a very common type of phishing technique. It is a threat that involves sending email messages with fake information on email headers. Because a spoofed email and regular mail are similar in many aspects, email spoofing takes advantage of these similarities. Attackers can customize the information in several fields such as "Return-Path", "Reply-To", "From", "Subject", "Date", and "To". The "Return Path"is where bounce messages go if the email fails to deliver. In legitimate emails, the "From"and "Return-Path"are typically consistent, representing the same source. However, in the case of email spoofing, there is often a discrepancy between these two fields. Scammers frequently manipulate the "From"address to appear as a trustworthy source, although they forget to modify the "Return-Path".
One of the main warning signs is an inconsistency between the "Reply-To"and "From"addresses. This disparity suggests the sender might be attempting to hide their true identity, which is a common phishing attempt approach. Additionally, if the "From" address does not align with the entity the email claims to represent, it further raises even more questions about the email's credibility. Another important detail is the nature of the "Subject"line. Phishing emails frequently have subject lines that are concerning, urgent, or too appealing in an attempt to get the receiver to act immediately without closely examining the legitimacy of the email. The "Date"field also needs to be taken into consideration. Attackers might use data that are not logical, including dates in the future or the past. Also, if the "To" address does not specifically name you, it can be indicative of phishing. Phishing emails often lack specific identification of the recipient, suggesting a broad targeting strategy known as mass mailings.

Another technical aspect of the email's metadata that can provide important information is the **spf!**. A "Fail"or "SoftFail"status from the **spf!** check, or a lack of **spf!** validation, raises serious questions about the legitimacy of the email. Also, another important point is that the IP address must line up with the sender's email service. If this does not happen, it suggests that the email may have been sent from an unauthorized or suspicious server. The "Received"fields can also provide crucial information, tracing the email's path across the internet. Unknown servers along this path, especially at the beginning or end, suggest that the email routing process may be compromised, raising the possibility of a phishing attempt.

Hijawi et al. [1] categorized the spam features into three primary groups based on the examination of the features present in the relevant studies in their literature: attachment features, payload (body) features, and header features. The header features were grouped into two classes, called email metadata and subject, and are displayed in Figure **??**.

ID	Feature Details	Type	Studies	ID	Feature Details	Type	Studies
1	Year	Metadata	[15]	26	Replay to MIL?	Metadata	[15]
2	Month	Metadata	[15]	27	Replay to Yahoo?	Metadata	[15]
3	Day	Metadata	[13] , [15]	28	Replay to AOL?	Metadata	[15]
4	Hour	Metadata	[13] , [15]	29	Replay to Gov?	Metadata	[15]
5	Minute	Metadata	[13] , [15]	30	X-Mailman-Version	Metadata	[15]
6	Second	Metadata	[13] , [15]	31	Exist Text/Plain?	Metadata	[15]
7	From Google?	Metadata	[15]	32	Exist Multipart/Mixed?	Metadata	[15]
8	From AOL?	Metadata	[15]	33	Exist Multipart/Alternative?	Metadata	[15]
9	From Gov?	Metadata	[15]	34	Number of characters.	Subject	[13]
10	From HTML?	Metadata	[15]	35	Number of capitalised words.	Subject	[13]
11	From MIL?	Metadata	[15]	36	Number of words in all uppercase.	Subject	[13]
12	From Yahoo?	Metadata	[15]	37	Number of words that are digits.	Subject	[13]
13	From Example?	Metadata	[15]	38	Number of words containing only letters.	Subject	[13]
14	To Hotmail?	Metadata	[15]	39	Number of words containing letters and number.	Subject	[13]
15	To Yahoo?	Metadata	[15]	40	Number of words that are single letters.	Subject	[13]
16	To Example?	Metadata	[15]	41	Number of words that are single digits.	Subject	[13]
17	To MSN?	Metadata	[15]	42	Number of words that are single characters.	Subject	[13]
18	To Localhost?	Metadata	[15]	43	Max ratio of uppercase letters to lowercase letters of each word.	Subject	[13]
19	To Google?	Metadata	[15]	44	Min of character diversity of each word.	Subject	[13]
20	To AOL?	Metadata	[15]	45	Max of ratio of uppercase letters to all characters of each word.	Subject	[13]
21	To Gov?	Metadata	[15]	46	Max of ratio of digit characters to all characters of each word.	Subject	[13]
22	To MIL?	Metadata	[15]	47	Max of ratio of non-alphanumeric characters to all characters of each word.	Subject	[13]
23	Count of “To” Email	Metadata	[13]	48	Max of the longest repeating character.	Subject	[13]
24	Replay to Google?	Metadata	[15]	49	Max of the character lengths of words.	Subject	[13]
25	Replay to Hotmail?	Metadata	[15]	-	-	-	-

Figura 2.6: Hijawi et al. [1] proposed header features.

Abadla et al. [14] in their study, used a dataset that has around 3800 records and 31 features related to the body of the email message, the subject box, and the sender's address. The proposal highlights specific characteristics often found in phishing emails, such as the inclusion of words like "urgent"and "suspension"in the subject line. Attackers deliberately use these terms to make victims frightened and force them to act right away. In addition, they identified that phishers use header phrases like "Fwd: mail"and "Re: mail"to create the sense of a continuing conversation, which increases the possibility that the receiver may interact with the email. This analysis of header features is crucial in understanding the linguistic and psychological strategies used in phishing attacks, thereby aiding in the development of more effective detection mechanisms. They introduced also the concept of “subject richness”, which pertains to the ratio of the number of words to the number of characters in the subject line. Turns out that this feature is crucial as it influences the open rate of an email.

In the context of phishing detection, the "Content-Type"field within attachments is a significant indicator. Phishing emails may contain attachments with content-types that are commonly associated with executable or scriptable content, such as "application/x-msdownload"for executables or "application/x-shockwave-flash"for Flash objects, which are potentially harmful. Moreover, attackers may disguise malicious attachments with a benign-looking content-type, such as "application/pdf"or "image/jpeg", while the actual file is a harmful executable. As mentioned in Caldwell [15] work, the attachment serves as the ideal camouflage for introducing an advanced persistent threat, commonly disguising itself as either a PDF file or a potentially flawed Office document, appearing subtle enough for people to accidentally execute.

In the Dewan et al. [16] study, a significant portion is devoted to analyzing the attachment names and types in spear phishing emails. The study highlights a clear distinction between the attachment names used in spear phishing and general spam or phishing emails. Spear phishing emails tend to have attachment names that appear more realistic and genuine, as opposed to the more irrelevant and lengthy names found in general spam or phishing emails.

This difference in naming conventions indicates that spear phishing attacks are crafted with more effort to make them appear legitimate and trustworthy, thereby increasing the likelihood of the recipient opening the attachment. Additionally, the paper discusses the types of file formats commonly used in these emails. Both spear phishing and general spam/phishing emails prominently feature executable (.exe, .bat, .com) and compressed (.rar, .zip, .7z) file types, along with common document formats like Microsoft Word, Excel, PowerPoint, and PDF files. The research explores the presence of attachments in the email body, indicating whether an email contains an attachment, and this feature, among others, is used to analyze and classify emails in their dataset. Although they do not use the attachment name and type as features for their classification, they can be used to enhance the detection of phishing emails.

The proposal presented by Li et al. [17] delves into the complexities of detecting phishing emails, particularly focusing on the role of email attachments in these malicious activities. In addressing this challenge, they propose an email feature extraction algorithm that focuses on various aspects of emails, including the names and suffixes of attachments. In the end, they got high accuracy results, which demonstrates the effectiveness of their approach, especially in the context of detecting phishing emails with attachments.

In the upcoming section of this thesis, the analysis of email body content will be discussed.

2.2 AI for Phishing Detection

As phishing techniques evolve and become increasingly sophisticated, traditional methods like rules-based filters and signature detection are no longer enough to keep us safe. This is where **ml!** (**ml!**) and **dl!** (**dl!**) come into play as powerful tools that can enhance our ability to detect phishing attacks. These artificial intelligence techniques enable systems to learn and adapt, allowing them to recognize subtle patterns and anomalies that may trick traditional detection approaches. By incorporating **ml!** and **dl!** into phishing detection, we not only aim to address the difficulties of identifying these deceitful communications but also play a crucial role in strengthening cybersecurity measures. This section will discuss the latest techniques for detecting and protecting against various email phishing attacks, including **ml!** and **dl!** models. In addition, it will address the field of **ai!** capable of automatically generating and understanding natural human languages, known as **nlp!**.

2.2.1 Natural Language Processing

As a branch of **ai!**, **nlp!** focuses on the interaction between computers and human language. It combines the power of linguistics and computer science to enable machines to understand, interpret, and respond to human language in a valuable and meaningful way. Human signs and languages, such as voice, writing, and text, can be automated with a certain level of accuracy using **nlp!** techniques [18].

The relevance of **nlp!** in detecting phishing emails is based on its ability to analyze and understand the textual content of emails. Phishing emails frequently include linguistic clues different from those in normal correspondence, and trick recipients into revealing sensitive

information. These cues can be subtle, such as the use of specific words or phrases, or more obvious, such as poor grammar and spelling. In either case, these linguistic features can be used to identify phishing emails and **nlp!** techniques can be used to extract and analyze them.

The **nlp!** field is vast and has a wide array of techniques, each contributing uniquely to the understanding and processing of language in the **nlp!** process. Data preprocessing is one step in **nlp!** that involves cleaning and transforming raw data into a format that can be understood and used by models. Some of the techniques used in this step include tokenization, normalization, stemming, the use of stop words, the application of regular expressions and both syntactic and semantic analysis. Feature extraction is another important step in **nlp!** that involves the extraction of features from the text. These features can be used to train **ml!** models to perform various tasks in this field. Techniques such as bag-of-words, word embeddings, and TF-IDF are used to extract features from text. After this, numerical features extracted by the previous techniques can be fed into **ml!** models. Depending on the task, different models can be used, such as classification, clustering, and regression models.

Vazhayil et al. [19] presents an insightful application of **nlp!** methods in conjunction with **ml!** models for phishing email detection. This study uses **tdm!** (**tdm!**) for the non-sequential representation of the corpus, followed by **svd!** (**svd!**) and **nmf!** (**nmf!**) to extract important features. These extracted features were then used to train various **ml!** algorithms, including **dt!** (**dt!**), **knn!** (**knn!**), **nb!** (**nb!**), **rf!** (**rf!**), **svm!** (**svm!**), and **lr!** (**lr!**). In the conclusion, they highlighted the effectiveness of this approach in distinguishing phishing emails from legitimate ones. However, the paper also acknowledges a limitation: the reliance on feature selection, which requires domain knowledge. To address this, future work could incorporate **dl!** models that can learn more complex patterns directly from the raw data, potentially improving efficacy. Gutierrez et al. [20] mentioned these types of defensive approaches frequently display a lack of adaptability. A major flaw with **nlp!** developed on **ml!** is their reliance on surface-level text analysis rather than exploring deeper semantics. This means that if different synonyms of words are chosen or the sentence construction is changed, it is difficult for **nlp!** built on **ml!** to analyze these changes.

Advancements in this area have led to the development of more sophisticated techniques. Unlike traditional **ml!** models, **dl!** models can automatically detect and learn features from raw text, allowing them to capture complex relationships between words and phrases in a language and to generalize to new and unseen examples. Some examples of **dl!** models that can be used in **nlp!** are **rnn!** (**rnn!**), **cnn!** (**cnn!**) and transformer models. **rnn!** is a type of neural network that can process sequential data, such as text, by using a hidden state to store information about previous inputs. **cnn!**, typically known for image processing, have also been effectively repurposed for **nlp!** tasks. Moreover, the emergence of Transformer models marks a significant leap. These models excel in understanding the context of language, processing each word concerning all other words in a sentence, and using self-attention mechanisms to capture the global relationships in a sentence.

2.2.2 Machine Learning approaches

The work proposed by Rabbi et al. [21] aims to find the most efficient techniques for preventing phishing attacks. For that, six **ml!** algorithms were separated including **lr!**, **knn!**, **ab!** (**ab!**), **mnb!** (**mnb!**), **gb!** (**gb!**), and **rf!**. One of the goals was to answer what is the most powerful **ml!** algorithm for detecting phishing emails and the results showed that the **rf!** performed better than other **ml!** algorithms having 98.38% accuracy and a low rate of false negatives. Although **rf!** obtained better results, its training time is relatively long when compared to others. However, the approach solely focuses on the email body features. There is more information such as the sender details, header information, and URLs in the email that can provide useful information for the model and increase performance.

In their study, the authors of [22] introduced a phishing URL detection method that integrates multiple **ml!** algorithms with unique hybrid features. These hybrid features are generated by first applying **pca!** (**pca!**) to word vector features, and then merging them with **nlp!** features. The dataset used in this study comprises approximately 37,000 URLs, evenly split between phishing and legitimate sites. Word vectors, also known as word embeddings, numerically represent words in a high-dimensional space. **pca!** is employed to reduce the dimensionality of these vectors. The resultant hybrid feature set, post-merging with **nlp!** features, encompasses 142 distinct features. Among the various **ml!** algorithms evaluated, the **rf!** algorithm exhibited the highest accuracy, achieving a remarkable 99.75%.

Shaukat et al. [23] proposed a solution that uses a three-layered approach to detect phishing websites. This multi-perspective layered evaluation has three layers: URL layer, text layer, and image layer. The first one analyzes URL features to detect phishing URLs, the second layer looks for spam content in website text by using **nlp!** and the last one categorizes the content of websites by processing text and graphics from advertising. The PhishTank dataset containing 20,000 phishing URLs and the SMS spam and ham dataset from Kaggle were used to train the machine learning models for the first two layers. The third layer takes the images from the websites as input to convert them into text so that they can be readable and given as input to the second layer model. For the URL classification the **dt!**, **rf!**, **mp!** (**mp!**), **svm!**, **lr!** and XG Boost models were tested. **nb!** and Linear **svc!** (**svc!**) models were used in the second layer to perform phishing text classification. The results showed up to 91.2% accuracy in the detection of legitimate or phishing URLs with XGBoost, and 98.9% accuracy with the Linear **svc!** model in the text analysis step.

Karhani et al. [24] presents a novel approach to detecting phishing URLs and SMS-based phishing (smishing) attacks. This approach combines domain-related features with **nlp!** techniques. The features related to domains are extracted and used alongside **nlp!**, which is trained on actual smishing messages, to detect attacks accurately. The study proposes integrating this detection system with the open-source **misp!** (**misp!**). This integration enhances the storage and utilization of flagged phishing domains. The dataset for this study includes data from TELUS Corporation and publicly available sources, featuring a mix of phishing and legitimate domains and SMS messages. The methodology involves a hybrid model that combines a **dt!** model and an **nlp!** model using **svc!**. The model demonstrates an

impressive accuracy of 99.40% and an F1 score exceeding 99%. The domain checker, part of the hybrid model, showed notable generalization capabilities with an F1 score of 99.01% and an accuracy of 98.04%. The **nlp!** checker, while effective, did not generalize as well to the confirmed phishing dataset provided by TELUS, with an F1 score and accuracy of 92.98% and 86.88% respectively. When both models were combined, the **nlp!** checker effectively corrected 69.35% of the domain checker's false negatives, improving the final accuracy to 99.40%.

2.2.3 Deep Learning approaches

The authors of [25] developed a phishing detection model focusing on the text of web pages rather than URL addresses. This model uses **nlp!** and **dl!** algorithms, specifically using the Keras Embedding Layer with **glove!** (**glove!**) to exploit semantic and syntactic features of webpage content. The method involves four phases: word parsing, data pre-processing, feature representation, and feature extraction. This approach ensures that important words and the order in which they appear are both considered for analysis. The model's performance was evaluated using four DL algorithms: **lstm!** (**lstm!**), **bilstm!** (**bilstm!**), **gru!** (**gru!**), and **bigru!** (**bigru!**). Notably, all four algorithms achieved a mean accuracy of at least 96.7%, with **bigru!** emerging as the top performer, achieving an accuracy of 97.39%. Further analysis revealed that both **gru!** and **bigru!** consistently outperformed **lstm!** and **bilstm!** in terms of test accuracy. Notably, **gru!** demonstrated the fastest training time, completing its training in just 240 seconds, which could be beneficial if rapid processing is required.

Alhogail e Alsabih [26] detailed an approach using **dl!** algorithms and **nlp!** to enhance phishing detection. **dl!** techniques are useful for phishing email detection because they perform especially well with unstructured data, like email content. The study highlights the advantages of **dl!** over other machine learning algorithms because traditional approaches require significant feature engineering. The core of this approach involves a **gcn!** (**gcn!**), which is described as a type of **cnn!** that operates directly on graphs. By utilizing a single large graph made from the entire email corpus, it converts the document classification problem into a node classification problem. This study uses **nlp!** on email body features alongside **dl!** algorithms using **gcn!** to build an effective phishing email detection classifier. The proposed classifier is built through three main phases: data collection and preparation, construction of the detection model using **dl! gcn!** algorithms and testing the classifier in a supervised approach using testing data for validation. This model demonstrates its effectiveness in detecting phishing emails based solely on body text, achieving an impressive accuracy rate of 98.2% with a low false-positive rate of 0.015.

The work presented by Fang et al. [27] introduces THEMIS, a new phishing email detection model, which utilizes an improved **rcnn!** (**rcnn!**) combined with a multilevel vector approach and an attention mechanism. The model analyzes the email structure, focusing on four detailed parts: the email header, the email body, and text at both the word and character levels. The THEMIS model, combining multilevel embedding with the improved **rcnn!**-Attention model, vectorizes the email's text structure and applies **bilstm!** for better email representation. The attention mechanism is applied between the email header and body, allowing the model to

prioritize more significant information from these parts. This complex approach enables THEMIS to perform exceptionally well on an unbalanced dataset, achieving an accuracy rate of 99.848% and a very low false positive rate (FPR) of 0.043%, which shows its effectiveness in accurately identifying phishing emails while minimizing the misclassification of legitimate emails.

Atawneh e Aljehani [28] proposal explores the use of **dl!** techniques, including **cnn!**, **lstm!** networks, **rnn!**, and **bert!** (**bert!**), for detecting email phishing attacks. The proposed model in the study involved collecting, preparing, and utilizing a dataset for training and testing the previously mentioned **dl!** models for phishing detection. This included steps like dataset acquisition, data preparation, feature extraction, and model training and testing. The research involved removing HTML tags, numbers, punctuations, stop words, and infrequent words from the email datasets. Stemming was also applied to reduce words to their base form. These preprocessing steps were essential for reducing noise in the data and enabling effective learning by the **dl!** models. All these models achieved high accuracy, with the best performance observed using **bert!** and **lstm!**, reaching an accuracy of 99.61%.

Table **??** summarizes the proposed works discussed in this section related to phishing detection. The literature review revealed several machine learning and deep learning techniques that can be used to detect phishing emails. It also evaluated the performance of various techniques in terms of accuracy. Complementing this analysis, the subsequent table, Table **??**, offers an in-depth overview of the datasets employed in these studies.

Authors	**Purpose**	**Major Themes**	**Accuracy**
Rabbi et al. [21]	Phishing emails detection	Using several **ml!** models	98.38%
Kumar [22]	Phishing URL detection	**ml!** models with hybrid features	99.75%
Shaukat et al. [23]	Phishing website detection	URL, text and image as input in several **ml!** models	98.90%
Karhani et al. [24]	Detecting phishing URL's and SMS-based attacks	Use of **ml!** models and integration with **misp!**	99.40%
Benavides-Astudillo et al. [25]	Phishing website detection based on text	Natural language and **dl!**	97.39%
Alhogail e Alsabih [26]	Email phishing detection and **dl!**	**nlp!** and graph conventional network	98.20%
Fang et al. [27]	Phishing email detection	Use of **rcnn!** with multilevel vectors and attention mechanism	99.84%
Atawneh e Aljehani [28]	Email phishing detection and **dl!**	Use of several **dl!** models: **cnn!**, **rnn!**, **lstm!**, and **bert!**	99.61%

Tabela 2.1: Summary of the discussed works.

Dataset	Description	Used by
TREC Public Coprus	Collection of email messages collected between April 8 and July 6, 2007. There are about 50,000 deceptive emails and about 25,000 legitimate.	[21]
Ling	The Ling-Spam dataset is a collection of 2,893 spam and non-spam messages curated from the Linguist List.	[21]
PhishTank	At PhishTank, the phishing data is reported by users and again tested by others to label them as phishing ones.	[23], [24]
SMS Spam Collection	Public set of SMS labelled messages, with 5,574 tagged (ham/spam).	[24]
The National University of Singapore SMS Corpus	This is a corpus of more than 67,000 SMS messages collected for research at the Department of Computer Science at the National University of Singapore.	[24]
PhishLoad	PhishLoad is a phishing database that contains HTML code, URL and another data relevant to phishing websites.	[25]
CLAIR collection	Collection of more than 2,500 fraud emails, dating from 1998 and later.	[26]
IWSPA-AP Corpus	PThe sources of the legitimate email include email collections from Wikileaks archives, such as the Democratic National Committee, Hacking Team, Sony emails, Enron Dataset, SpamAssassin, etc.	[27]
Enron Dataset	It is a dataset that contains about 50,000 spam and 43,000 ham emails. **dl!**	[28]

Tabela 2.2: Datasets used in the discussed works.

Table **??** presents a summary of recent works by various authors, the datasets employed, and the corresponding evaluation methods utilized to assess the performance of their proposed classifiers. A common evaluation method observed is the train-test split, with proportions ranging from 80%/20% to 70%/30%. This method provides a straightforward approach to assess the model on unseen data, ensuring that the classifier can effectively detect phishing attempts beyond the data it was trained on.

K-fold cross-validation, where K varies from 3 to 10, is another prevalent evaluation technique noted in the table. Cross-validation is a robust statistical method that maximizes the use of available data by rotating the train-test sets, thereby offering a comprehensive assessment of the classifier's performance. It suggests a consistency in the methodology that enables a fair comparison across different studies, as it mitigates the impact of any potential biases inherent in a single train-test split.

Authors	Datasets	Classifier's Evaluation Method
Rabbi et al. [21]	TREC Public Corpus and Ling	Train test split (80%/20%)
Kumar [22]	-	Train test split
Shaukat et al. [23]	PhishTank	Train test split
Karhani et al. [24]	PhishTank, SMS Spam Collection and University of Singapore SMS Corpus	K-fold Cross-validation (K=5)
Benavides-Astudillo et al. [25]	PhishLoad	K-fold Cross-validation (K=5)
Alhogail e Alsabih [26]	CLAIR Collection	Train test split (70%/30%) and K-fold Cross-validation (K=3)
Fang et al. [27]	IWSPA-AP Corpus	K-fold Cross Validation (K=10)
Atawneh e Aljehani [28]	Enron Dataset	Train test split (70%/30%)

Tabela 2.3: Comparative Overview of Datasets and Classifier Evaluation Methods in Phishing Email Detection Research.

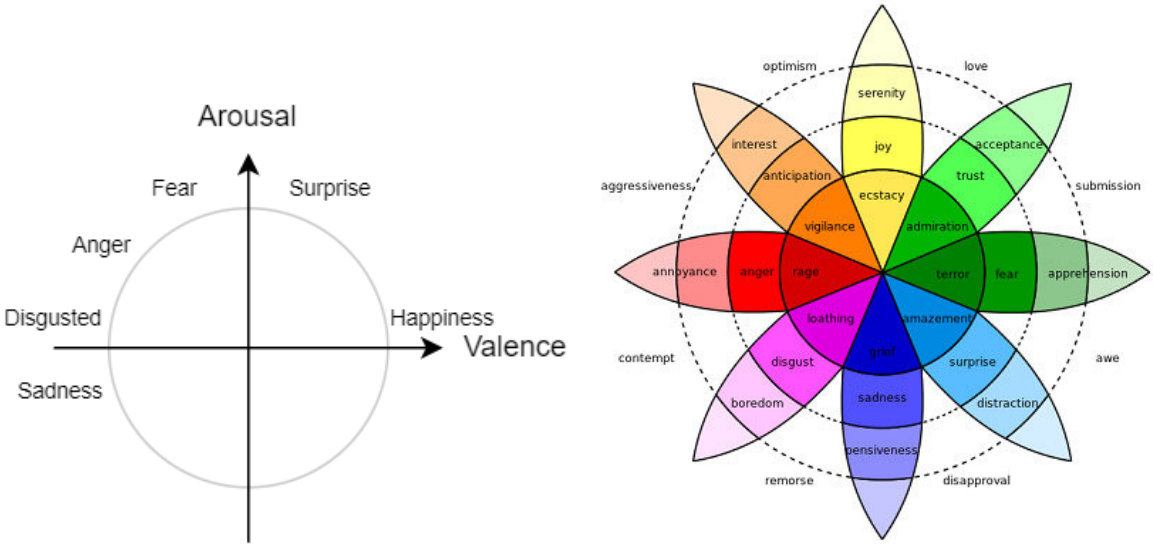
2.3 Sentiment analysis

Sentiment analysis is a **nlp!** technique that refers to the process of evaluating and determining the sentiment, which is characterized as feeling or emotion, contained in a certain text. The core of sentiment analysis lies in polarity detection, which classifies text into basic categories like positive, negative, or neutral. However, sentiment analysis goes beyond polarity to identify particular emotions like happiness, frustration, anger, and sadness.

Emotion models in psychology are broadly classified into two categories: dimensional and categorical models. The first one represents emotions based on three parameters - valence, arousal, and power. Valence refers to the polarity of the emotion, arousal refers to the intensity of the emotion, and power refers to the degree of control over the emotion. The second model, on the other hand, classifies emotions into discrete categories, such as happiness, sadness, anger, and fear. Depending on the categorical model, emotions can be categorized into several different categories [29]. Ekman's and Plutchik's models are two examples of categorical and dimensional models, respectively. Ekman's model classifies emotions into six categories: anger, disgust, fear, happiness, sadness, and surprise, whereas Plutchik's model includes a wider range, offering a dimensional representation of emotions. Both models can be seen in Figure **??**.

Generally, the input to a sentiment classification model is a piece of text, and the output is the probability of a certain sentiment or emotion. Typically, this probability is based on either hand-generated features, word n-grams, TF-IDF features, or using **dl!** models to capture sequential long- and short-term dependencies. Many emotion systems use lexicons, which are lists of words and their corresponding emotions. These lexicons can be used to determine the sentiment of a text by counting the number of words that match the words in the lexicon. However, this approach is limited by the fact that it does not consider the context of the words, which can lead to inaccurate results.

Despite its wide applications, sentiment analysis faces several challenges. One of the



(a) Ekman's model. **(b)** Plutchik's model.

Figura 2.7: Emotions models.

most significant is detecting sarcasm and irony, as these often convey the opposite meaning of the literal words used, leading to potential misinterpretation. Additionally, sentiment analysis must contend with contextual and cultural variations. The same phrase may carry different meanings in different cultures or situations, complicating universal model applicability. Moreover, because human language can be unclear and different people might see it differently, sentiment analysis becomes more complicated. What is considered a positive sentiment in one context may be neutral or even negative in another, which is why we need advanced models that understand the context.

The Sailunaz e Alhajj [30] study provides a robust example of an integrated approach. The researchers focused on extracting sentiment and emotion from tweets and replies on specific topics. This involved creating a dataset encompassing text, user emotion, sentiment information, and various other parameters. The text within was labeled with polarity information using 'Positive', 'Negative' and 'Neutral' and with specific emotions 'Anger', 'Disgust', 'Fear', 'Joy', 'Sadness', 'Surprise' and 'Neutral', known as Ekman's emotion model. The researchers used three different **ml!** models: **nb!**, **svm!** and **rf!**. The **nb!** model achieved the highest accuracy in both the sentiment and emotion classification tasks, being 66.9% and 47.3% respectively.

A specific example is the Sentiment Analysis Module detailed in the Sathish et al. [31] study, where this module captures the emotions or sentiments expressed in emails. It employs the **nltk!** (**nltk!**), an open-source Python library, to analyze the text based on common and repetitive sentiment words included in the training set. Determining sentence polarity is an important part of this module since it helps to comprehend the emotional tone that an email provides. The system pre-determines the polarity of specific polar words to interpret the sentiment accurately.

Models	Datasets	Emotions model	Limitations
rf!, **svm!** and **nb!** [30]	Collection of tweets	Ekman's model	Use of tweets instead of emails
nltk! and **nb!**, **dt!** [31]	Collection of emails	-	Not the ideal emotions selected
RoBERTa **liu2019roberta**	GoEmotions **demszky2020goemotions**	Emotions collected from Ekman's, Plutchik's and other models	Only supports English language
DistilBERT **sanh2019distilbert** Multilingual	Multilingual-sentiment dataset	-	Sentiment analysis instead of emotions

Tabela 2.4: Comparative Overview of Datasets and Classifier Evaluation Methods in Phishing Email Detection Research.

2.4 Insights

The analysis of phishing email detection strategies and tactics covered in this chapter provides a comprehensive overview of the state of the art at the moment.

One notable development in the phishing detection field is the role of **ai!**, particularly **nlp!** and **ml!**/**dl!** approaches. **nlp!**'s capacity to analyze and evaluate email language is very helpful in identifying phishing emails, as they frequently have linguistic indicators that set them apart from legitimate correspondence. Although **ml!** models have been used to detect phishing emails, they are limited in their ability to learn complex patterns from raw data. This is where **dl!** models come in, as they can learn more complex patterns directly from the raw data, potentially improving efficacy.

The quality and diversity of datasets used in training these models also demand attention. A model's performance is only as good as the data it's trained on. Datasets that lack diversity or representativeness can lead to models that are ineffective in real-world scenarios, where phishing tactics are continually evolving. This is particularly pertinent given the dynamic nature of phishing, where detection systems must not only be technologically advanced but also strategically adaptable to counteract new and evolving threats.

Based on the review of the state-of-the-art in phishing email detection and sentiment analysis, it is evident that despite notable progress in detecting and mitigating phishing attacks via machine learning and deep learning techniques, there is still a notable gap in integrating sentiment analysis into phishing detection systems. This process is not merely about polarity detection (classifying text as positive, negative, or neutral) but extends to identifying specific emotions like happiness, frustration, anger, and sadness. Existing phishing detection systems do not perform this type of analysis, which is crucial in understanding the psychological strategies used in the attacks.

In addition, the dynamic nature of phishing techniques needs continual learning and adaptation in detection systems. Phishing tactics evolve regularly, therefore detection systems

need to be able to pick up on these changes and adjust accordingly. This adaptive approach is not just a technological improvement but a strategic necessity in the ongoing battle against these cyber threats.

CAPÍTULO 3

Methodology

3.1 Work Plan

In this section is presented the work plan for this thesis. In Figure **??** is displayed a Gantt chart which maps the timeline of each task in the development process. The purpose of this work plan is to ensure a systematic approach to the development and writing of this thesis. It reflects a thoughtful and comprehensive approach to tackling a complex and dynamic problem in cybersecurity. The work plan is divided into three main phases: *Research*, *Development* and *Writing*, not being treated separately but rather as a continuous process.

The plan spans over ten months, with each task carefully allocated to specific weeks within these months, enabling a clear roadmap for the thesis progression. The next steps are described in more detail:

- **Research:** To finish this phase, it is necessary to complete the architecture planning and the models to be used;
- **Development:** In this phase, the development of the framework will be carried out, as well as the tests and validation of the results. This involves two main tasks: Phishing detection and sentiment analysis module development. Testing the model with real data and identifying possible limitations are addressed in this phase;
- **Writing:** The writing of the thesis will be done in parallel with the development of the framework so that the writing of the thesis is not left to the end. This includes the writing of the proposal chapter as well as the completion of the rest of the thesis.

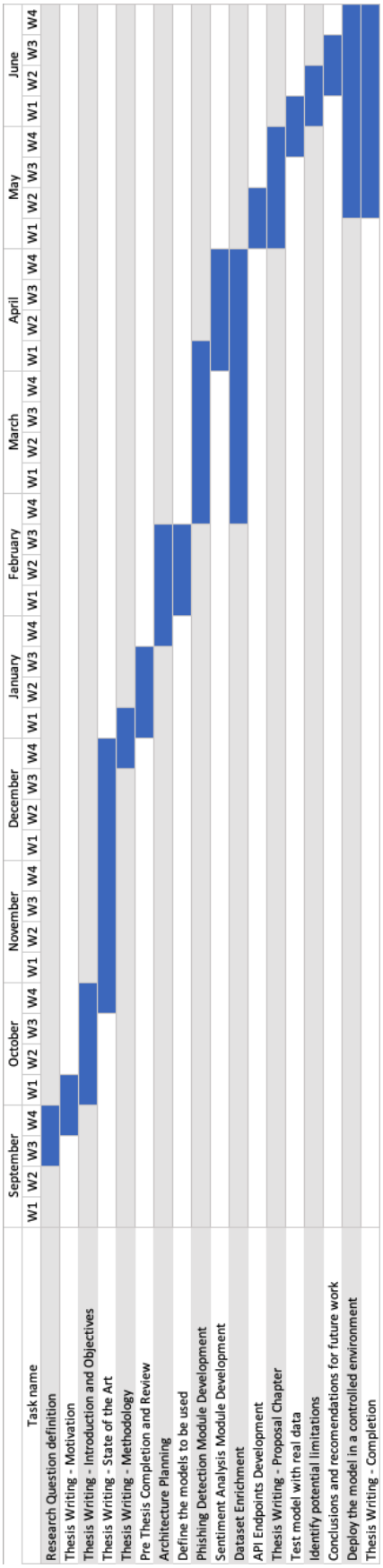


Figura 3.1: Gantt diagram containing the tasks that will be developed during this thesis work.

Referências

[1] W. Hijawi, H. Faris, J. Alqatawna, A. M. Al-Zoubi e I. Aljarah, «Improving email spam detection using content based feature engineering approach,» em *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2017, pp. 1–6. DOI: 10.1109/AEECT.2017.8257764.

[2] Y. Li e Q. Liu, «A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments,» *Energy Reports*, vol. 7, pp. 8176–8186, 2021. DOI: 10.1016/j.egyr.2021.08.126.

[3] A. Bendovschi, «Cyber-attacks–trends, patterns and security countermeasures,» *Procedia Economics and Finance*, vol. 28, pp. 24–31, 2015. DOI: https://doi.org/10.1016/S2212-5671(15)01077-1.

[4] «CISA - Malware, Phishing, and Ransomware,» CISA, 2023. URL: https://www.cisa.gov/topics/cyber-threats-and-advisories/malware-phishing-and-ransomware.

[5] K. D. Tandale e S. N. Pawar, «Different types of phishing attacks and detection techniques: A review,» em *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, IEEE, 2020, pp. 295–299. DOI: 10.1109/ICSIDEMPC49020.2020.9299624.

[6] «Phishing Activity Trends Report, 4rd Quarter 2022,» APWG, 2022. URL: https://docs.apwg.org/reports/apwg_trends_report_q4_2022.pdf.

[7] «Phishing Activity Trends Report, 3rd Quarter 2020,» APWG, 2020. URL: https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf.

[8] «ENISA Threat Landscape 2020 - Phishing,» ENISA, 2020. URL: https://www.enisa.europa.eu/topics/cyber-threats/threats-and-trends/etl-review-folder/etl2020-phishing.

[9] C. Dürscheid, C. Frehner, S. C. Herring, D. Stein e T. Virtanen, «Email communication,» *Handbooks of Pragmatics [HOPS]*, n.º 9, pp. 35–54, 2013. DOI: 10.1515/9783110214468.35.

[10] M. Vacek, «How to survive email,» em *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2014, pp. 49–54. DOI: 10.1109/SACI.2014.6840097.

[11] F. Kooti, L. M. Aiello, M. Grbovic, K. Lerman e A. Mantrach, «Evolution of conversations in the age of email overload,» em *Proceedings of the 24th international conference on world wide web*, 2015, pp. 603–613. DOI: 10.1145/2736277.2741130.

[12] D. J. C. Klensin, *Simple Mail Transfer Protocol*, RFC 5321, out. de 2008. DOI: 10.17487/RFC5321. URL: https://www.rfc-editor.org/info/rfc5321.

[13] P. Resnick, *Internet Message Format*, RFC 5322, out. de 2008. DOI: 10.17487/RFC5322. URL: https://www.rfc-editor.org/info/rfc5322.

[14] R. Abadla, A. Alseiari, A. Alheili, M. S. Daoud e H. M. Al-Mimi, «Intelligent Phishing Email Detection with Multi-Feature Analysis (IPED-MFA),» 2023, pp. 12–18. DOI: 10.1109/ICCNS58795.2023.10193714.

[15] T. Caldwell, «Spear-phishing: how to spot and mitigate the menace,» *Computer Fraud & Security*, vol. 2013, n.º 1, pp. 11–16, 2013. DOI: https://doi.org/10.1016/S1361-3723(13)70007-1.

[16] P. Dewan, A. Kashyap e P. Kumaraguru, «Analyzing social and stylometric features to identify spear phishing emails,» em *2014 apwg symposium on electronic crime research (ecrime)*, IEEE, 2014, pp. 1–13. DOI: 10.1109/ECRIME.2014.6963160.

[17] Q. Li, M. Cheng, J. Wang e B. Sun, «LSTM based phishing detection for big email data,» *IEEE transactions on big data*, vol. 8, n.º 1, pp. 278–288, 2020. DOI: 10.1109/TBDATA.2020.2978915.

[18] C. Sathish, A. Mahesh, N. S. Karpagam, R. Vasugi, J. Indumathi e T. Kanchana, «Intelligent Email Automation Analysis Driving through Natural Language Processing (NLP),» 2023, pp. 1612–1616. DOI: 10.1109/ICEARS56392.2023.10085351.

[19] A. Vazhayil, N. Harikrishnan, R. Vinayakumar e K. Soman, «PED-ML: Phishing email detection using classical machine learning techniques CENSec@Amrita,» vol. 2124, 2018, pp. 69–76.

[20] C. N. Gutierrez, T. Kim, R. D. Corte et al., «Learning from the ones that got away: Detecting new forms of phishing attacks,» *IEEE Transactions on Dependable and Secure Computing*, vol. 15, n.º 6, pp. 988–1001, 2018. DOI: 10.1109/TDSC.2018.2864993.

[21] M. F. Rabbi, A. I. Champa e M. F. Zibran, «Phishy? Detecting Phishing Emails Using ML and NLP,» em *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, IEEE, 2023, pp. 77–83. DOI: 10.1109/SERA57763.2023.10197758.

[22] J. Kumar, «Hybrid Feature-Based Machine Learning Method for Phishing URL Detection,» Cited by: 0, 2023, pp. 222–227. DOI: 10.1109/ICSCCC58608.2023.10176901.

[23] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri e J. Xie, «A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning,» *Sensors*, vol. 23, n.º 19, 2023, Cited by: 0; All Open Access, Gold Open Access. DOI: 10.3390/s23198070.

[24] H. E. Karhani, R. A. Jamal, Y. B. Samra, I. H. Elhajj e A. Kayssi, «Phishing and Smishing Detection Using Machine Learning,» Cited by: 0, 2023, pp. 206–211. DOI: 10.1109/CSR57506.2023.10224954.

[25] E. Benavides-Astudillo, W. Fuertes, S. Sanchez-Gordon, D. Nuñez-Agurto e G. Rodríguez-Galán, «A Phishing-Attack-Detection Model Using Natural Language Processing and Deep Learning,» *Applied Sciences (Switzerland)*, vol. 13, n.º 9, 2023, Cited by: 2; All Open Access, Gold Open Access. DOI: 10.3390/app13095275.

[26] A. Alhogail e A. Alsabih, «Applying machine learning and natural language processing to detect phishing email,» *Computers & Security*, vol. 110, p. 102 414, 2021, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102414.

[27] Y. Fang, C. Zhang, C. Huang, L. Liu e Y. Yang, «Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism,» *IEEE Access*, vol. 7, pp. 56 329–56 340, 2019. DOI: 10.1109/ACCESS.2019.2913705.

[28] S. Atawneh e H. Aljehani, «Phishing Email Detection Model Using Deep Learning,» *Electronics*, vol. 12, n.º 20, p. 4261, 2023. DOI: https://doi.org/10.3390/electronics12204261.

[29] P. Nandwani e R. Verma, «A review on sentiment analysis and emotion detection from text,» *Social Network Analysis and Mining*, vol. 11, n.º 1, p. 81, 2021. DOI: 10.1007/s13278-021-00776-6.

[30] K. Sailunaz e R. Alhajj, «Emotion and sentiment analysis from Twitter text,» *Journal of Computational Science*, vol. 36, p. 101 003, 2019, ISSN: 1877-7503. DOI: https://doi.org/10.1016/j.jocs.2019.05.009.

[31] C. Sathish, A. Mahesh, N. S. Karpagam, R. Vasugi, J. Indumathi e T. Kanchana, «Intelligent Email Automation Analysis Driving through Natural Language Processing (NLP),» em *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, 2023, pp. 1612–1616. DOI: 10.1109/ICEARS56392.2023.10085351.