esprit
Se former autrement

HONORIS UNITED UNIVERSITIES

DATA SCIENCE PROJECT REPORT

Automatic Phishing Emails Detection Using Deep Learning

Élaboré Par

EMNA KHEMIRI
INES SALEM
IMEN METTICHI
SAMAR BELARBI
MED AMINE AMDOUNI
SEIF SAHNOUN
MED AMINE SALEM ZITOUN

Framed by : Mr.Abderrahmen BENAROUS
Mrs. Ghada SETTI

University Year : 2022-2023

Acknowledgment

We would like to express our sincere gratitude and appreciation to Mr. Abderrahmen Benarous and Ms. Dorra Trabelsi for their invaluable guidance, support, and expertise throughout the duration of this project. Their extensive knowledge and deep understanding of deep learning approaches have been instrumental in shaping the direction and success of our research.

Mr. Abderrahmen Benarous' insightful ideas, constant support, and guidance have been critical to the success of our initiative. His knowledge in cybersecurity and data science has been crucial in providing us with guidance and direction, ensuring that our work stayed focused and aligned with industry best practices.

Finally, we would like to extend our most sincere thanks to the members of the jury that provided us with constructive feedback and helped us improve our work.

Abstract

Phishing has emerged as a predominant threat to Internet users, making it difficult to ensure security in the world of communication. Phishing attempts aim to illegally obtain consumers' private information, such as passwords, credit card numbers, and account login information. Attackers utilize duplicitous methods to conceal their intentions, such as by constructing websites and sending email messages that closely resemble trustworthy sources. Thus, unwary users run the risk of unintentionally disclosing their private information and becoming victims of identity theft and financial loss. That is why it is imperative to accurately identify these malicious attempts and stop them before they cause any harm.

In this project, we utilized deep learning approaches including NLP (Natural Language Processing) and Neural Networks. Additionally, we incorporated cybersecurity techniques such as DNS analysis and attachment malware protection tools to develop an efficient phishing E-mail detection system.

Table des matières

Table des figures

General Introduction

Phishing attacks present a serious risk to both individuals and companies since they can result in monetary losses, data breaches, and compromised security. A major problem in maintaining a secure digital environment is seeing and avoiding phishing emails. In this technical report, we outline a project that aims to create a deep learning-based system for detecting phishing emails.

The goal of this project is to develop an automated system that can recognize and categorize phishing emails with accuracy, adding an extra line of defense against online threats. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) with Gated Recurrent Units (GRU) or Long Short-Term Memory (LSTM), as well as supplementary methods like sender domain analysis and attachment verification, are deep learning algorithms that we will use to enhance the detection modelâs precision and effectiveness.

Phishing emails frequently use sophisticated tactics to trick receivers, impersonating reliable sources and persuading people to divulge critical information or take destructive actions. Traditional rule-based strategies and pattern-matching techniques find it difficult to keep up with how phishing attempts are constantly developing. By automatically discovering and extracting complex patterns and features from huge datasets, deep learning algorithms have demonstrated promise in solving this problem.

Preprocessing the email data, collecting key features from the email text, and training a deep learning model on a labeled dataset of phishing and legitimate emails comprise our approach. CNN and LSTM layers are implemented to assess the phishing patterns in the content of the email, while RNN layers with GRU cells are utilized for analyzing the validity of the URLs. In addition, we incorporate sender domain analysis and an attachment malware verification tool to determine the legitimacy of the

e-mail.

The anticipated outcome of this project is a phishing email detection system that can reliably identify and classify suspicious or malicious emails while minimizing false positives and false negatives. This technology can help corporations reinforce their email security infrastructure, secure sensitive data, and mitigate potential hazards associated with phishing attempts.

Chapter 1

Business Understanding

1.1 Introduction

This section focuses on developing a thorough understanding of phishing emails, including their features, effect, and obstacles. We can better understand the developing landscape of this cybersecurity threat by investigating the reasons behind phishing attempts and evaluating typical strategies utilized by attackers. In addition, we will discuss the dangers of falling prey to phishing emails, as well as the possible penalties for people and corporations.

1.2 Cybersecurity

Cybersecurity, an interdisciplinary field that spans various practices, processes, and technologies, aims to protect computer systems, networks, devices, and data from unwanted access, theft, damage, or disruption through diverse practices, methods, and technologies. It includes network security, information security, data security, application security, cloud security, mobile security, and endpoint security. The reliance on technology and the pervasiveness of the internet in today's digital landscape highlight the importance of cybersecurity. Its role is critical in protecting organizations and individuals from the negative effects of cybercrime. Cybersecurity experts work to combat hostile activities such as viruses, malware, phishing assaults, and social engineering by creating strong security protocols, rules, and processes, as well as cutting-edge tools and technologies.

1.3 Phishing

1.3.1 Definition

Phishing is the practice of online criminals impersonating trustworthy sites or organizations to trick unwary users into disclosing their private information. Most phishing attempts use social engineering techniques, in which the targets are persuaded to experience emotions like urgency, worry, or curiosity. This manipulates readers into responding right away without checking the emailâs validity by clicking on a link, opening an attachment, or entering personal information on a fake website. As a result, attackers can obtain sensitive data such as login credentials, credit card numbers, or personal information.

FIGURE 1.1 – The process of e-mail phishing

1.3.2 Types of phishing

There are two main types of phishing emails : 1. Spoofed phishing : In this type of phishing, the attacker uses a fake sender email address that is designed to look like it’s from a trustworthy entity. The email may contain a link to a fake website that looks like a legitimate one.
2. Spear phishing : In this type of phishing, the attacker targets a specific individual or organization and tailors the phishing email to them. The attacker may gather information about the target to make the email more convincing. Spear phishing is often more sophisticated and successful than spoofed phishing, as the attacker can create a more convincing email that is tailored to the target.

FIGURE 1.2 – An example of a spoofed phishing e-mail

1.3.3 Dangers of Phishing

Phishing emails have evolved as a widespread and highly dangerous cyber threat, needing a thorough awareness of their dangers. These false emails, designed by fraudsters to appear to be from reputable sources, pose major hazards to both individuals and companies. Phishing attacks accounted for more than 80 percent of reported security issues in 2020, according to current statistics (Source : Verizon's 2021 Data Breach Investigations Report). Financial loss, identity theft, reputational damage, and business disruption may all arise from falling victim to phishing (Source : Anti-Phishing Working Group, 2020). Phishing crimes can also result in data breaches within firms, exposing intellectual property, customer information, and sensitive corporate data. The 2014 phishing attack on Sony Pictures Entertainment, for example, exposed a massive quantity of personal material, including unreleased films, employee records, and executive communications (Source : The New York Times).

1.4 Requirements

1.4.1 Functional requirements

- **Email Analysis** : The system should be able to analyze the content of received emails to identify potential phishing attempts.
- **Phishing Detection** : The system should employ natural language processing (NLP) techniques and pattern recognition algorithm.
- **Phishing Alerts** : The system should provide a mechanism to report identified phishing emails to users or administrators for further action.
- **Real-time Detection** : The system should be capable of assessing in-

coming emails in real-time to promptly identify and respond to phishing attempts.
- **Gmail integration** : The system should integrate with the Gmail platform to analyze and detect phishing emails within the user's Gmail account.

1.4.2 Non-Functional requirements

- **Scalability** : The system should be able to efficiently manage an enormous flow of incoming emails, ensuring that the detection process is not hampered by email overload.
- **Performance** : The phishing detection model should run with minimal delay.
- **Data Protection** : The system should follow data privacy and security requirements, shielding sensitive information within emails while maintaining user confidentiality.
- **Security Compliance** : The system should be kept up to date with the current security standards and best practices.

1.5 Objectives

1.5.1 Business objectives

- **Improve Email Security** : Our primary business goal is to improve email security by identifying and mitigating phishing attempts, hence diminishing the risk of financial loss, data breaches, and reputational harm for individuals and businesses.
- **Reduce the impact of phishing attacks** : Minimize the effect of phishing attacks on individuals and organizations by recognizing and blocking phishing emails as soon as possible, thus alleviating potential financial and reputational damages.
Additional Sustainable Business Objectives aligned with the United Nations Sustainable Development Goals (SDGs) :
- **Industry, Innovation, and Infrastructure** : Encourage the creation and adoption of new technology and infrastructure solutions that improve email security and protect users from phishing attempts. This goal contributes to the development of resilient infrastructure and the growth of technology to enable sustainable development.
- **Peace, Justice, and Strong Institutions** : Help establish a safer and more secure digital environment by preventing cybercrime. This goal is consistent with the goal of fostering peaceful and inclusive communities,

guaranteeing access to justice, and establishing effective and responsible institutions at all levels.

1.5.2 Data Science objectives

- **Text-based Phishing Email Detection** : Deep learning techniques, notably CNN and LSTM, should be used to create a model capable of investigating email text and precisely recognizing phishing attempts based on patterns and features associated with phishing emails.
- **URL-based Phishing Detection** : Employ RNN models to analyze URLs and verify their maliciousness.
- **DNS lookup** : Run a DNS lookup on the sender's email address to confirm its authenticity and detect any suspicious or malicious websites associated with the email.
- **Attachment analysis** : Leverage the Cloudmersive API or similar solutions for email attachment evaluation, malware detection, and user protection from harmful files and documents.

1.6 Metrics and KPIs

It's crucial to evaluate the performance of the phishing detection algorithms to ensure we can compare them and choose the one that performs optimally in any given case. We will use : - **True Positive (TP)** : This is the proportion of phishing emails in the training dataset that a phishing detection algorithm properly classifies. Formally, the formula for TP is as follows : Where P is the number of phishing emails in the dataset, and NP denotes the number of phishing emails successfully categorized by the phishing detection model. TP = NP/P
- **True Negative (TN)** : This reflects the proportion of real emails that phishing detection models properly identify as such. The formula for TN is as follows if we represent the total number of legitimate emails as L and the number of legitimate emails that are accurately categorized as legitimate as NL : TN = NL/L.
- **False Positive (FP)** : This is the proportion of legitimate emails that are mistakenly labeled as phishing emails by the model. The FP formula is given like so if we represent the total number of legitimate emails as L and the number of legitimate emails that are mistakenly labeled as phishing as Nf : FP =Nf/L.
- **False Negative (FN)** : This is the proportion of phishing emails that

a phishing detection algorithm misclassifies as legitimate. The formula for FN is as follows if we represent the total number of phishing emails in the dataset as P and the number of phishing emails that the algorithm has determined to be genuine. FN = Npl/P.

Using TP, TN, FP, and FN, the accuracy can be computed as follows :
- **Accuracy** : It reflects the average number of emails that were correctly classified throughout the whole dataset using the following formula : Accuracy = (TP + TN)/ (TP + FP + FN + TN).
- **User reporting rate** : (Number of phishing emails reported by users) / (Total number of phishing emails received by users).

1.7 Conclusion

In conclusion, for organizations to properly handle this cybersecurity threat, a detailed understanding of phishing emails is essential. Organizations can boost their defenses and reduce the dangers associated with phishing emails by putting a priority on education, establishing strong security measures, and utilizing sophisticated detection technologies. This will eventually result in a safer digital environment for all stakeholders.

Chapter 2

Data Understanding

2.1 Introduction

An essential element in creating a reliable phishing email detection system is data understanding. Defining the features and content of the data that will be utilized to train and test the detection model is a requirement of this step. We can make wise selections and create suitable algorithms and models for efficiently identifying phishing emails by attaining insights into the data.

2.2 Phishing e-mails data collection

For this project, a broad dataset of phishing and legitimate emails was collected to train the model. The fraudulent data was acquired by extracting phishing emails from the publicly available "emails-phishing-Nazario" dataset, which was generated by researcher Jose Nazario. This collection contains 4290 real-world phishing emails that have already been discovered and recorded. Nevertheless, the Enron public legal emails dataset was used to extract legitimate emails. This data set contains a huge number of emails from the former energy business Enron Corporation ranging around 20k samples. The Enron data frame contains legitimate emails that can be used to train and test the phishing email detection algorithm. We developed a comprehensive dataset by combining these two datasets, which includes both phishing and legitimate emails. The dataset was meticulously tagged to distinguish between phishing and non-phishing emails, allowing supervised learning to efficiently train the detection model.

2.3 Phishing URLs Web Scraping

To improve the phishing email detection system, we acquired a complete dataset of phishing URLs in addition to valid and phishing emails. Two major sources were used in the data collecting process : a web scrape of the Phishtank website and a publicly accessible dataset from Kaggle. We used web scraping methods in order to further expand the dataset thus improving the accuracy of the model. This had provided us over 200k fraudulent URL besides the 450,000 URLs present in the Kaggle database.

```
# ### WEB SCRAPING NO NEED TO RUN IT AGAIN THE SCRAPED DATA IS SAVED TO A CSV FILE


# import requests
# from bs4 import BeautifulSoup
# import csv
# import time
# from concurrent.futures import ThreadPoolExecutor

# def scrape_page(page):
#     session = requests.Session()
#     url = f'https://phishtank.org/phish_search.php?page={page}&valid=y'
#     try:
#         response = session.get(url)
#         response.raise_for_status()
#     except requests.exceptions.RequestException as e:
#         print(f"An error occurred while fetching the URL: {e}")
#         return []

#     soup = BeautifulSoup(response.content, 'html.parser')
#     rows = soup.find_all('tr')[1:]
#     phish_data = []

#     for row in rows:
#         columns = row.find_all('td')
#         phish_url = columns[1].text.strip()
#         added_index = phish_url.find("added")
#         if added_index != -1:
#             phish_url = phish_url[:added_index].strip()
#         phish_data.append(phish_url)

#     return phish_data

# def main():
#     num_pages = 10000
#     all_phish_data = []

#     # Use ThreadPoolExecutor to parallelize requests
#     with ThreadPoolExecutor(max_workers=10) as executor:
#         for phish_data in executor.map(scrape_page, range(1, num_pages + 1)):
#             all_phish_data.extend(phish_data)
#             print(f"Scraped {len(phish_data)} URLs")

#     # Save the scraped data to a CSV file
#     with open('phishing_urls.csv', 'w', newline='', encoding='utf-8') as csvfile:
#         csvwriter = csv.writer(csvfile)
#         csvwriter.writerow(['URL'])
#         for url in all_phish_data:
#             csvwriter.writerow([url])

# if __name__ == "__main__":
#     main()
```

FIGURE 2.1 – Web scraping implemented algorithm.

2.4 Conclusion

The data collection phase was critical for generating diverse and informative datasets for the phishing email detection development process. The incorporation of the "emails-phishing-Nazario" dataset, as well as the Enron public legal emails dataset, provides an extensive range of phishing and valid emails. Furthermore, collecting phishing URLs from Kaggle and web scraping the Phishtank website enhanced the dataset's coverage and improved the system's capacity to reliably identify phishing attempts. We have created a solid basis for the project's upcoming phases by collecting these datasets. The information gathered will allow us to develop a strong and dependable phishing email detection system capable of discriminating between phishing and non-phishing emails.

Chapter 3

Data Preprocessing

3.1 Introdcution

In the context of a phishing email detection system, data preparation is a critical step in preparing data for analysis and modeling. It entails processing raw data in order to improve its quality, consistency, and usefulness. We can improve the detection system's performance and accuracy by addressing difficulties such as missing values, data cleaning, feature selection, and text preparation. Data preparation is essential for guaranteeing the data's quality and usefulness for later analysis and modeling phases.

3.2 E-mails dataset preprocessing

3.2.1 Dataset construction

We used two email datasets in this project : the "emails-phishing-Nazario" dataset and the Enron public legal emails dataset. We retrieved essential information such as the sender, topic, and content of each email from these databases. We classified the data to help with training by assigning a value of 1 to phishing emails and 0 to benign emails. During training, this labeling allows the model to learn patterns and differentiate between phishing and non-phishing cases.

	from	subject	content	class
1	NaN	You Have (1) Urgent Message From USAA Bank. Ac...	td>\n\n\n\n\n\n\n\n\n\n\nSecurity Preference...	1
2	NaN	Re: Important Notification	Dear Customer,\nTo learn how USAA protects you...	1
3	NaN	RE: HELP CENTER!	NaN	1
4	NaN	Account Notification	Dear Customer,\n\n\n\nWe emailed you a little ...	1

FIGURE 3.1 – Phishing emails dataset

	from	subject	content	class
4996	dana.davis@enron.com	Industrial Sector	Brad this file is also located in the O: drive...	0
4997	dana.davis@enron.com	Re: Calendar	---------- Forwarded by Dana Davis...	0
4998	dana.davis@enron.com	Re:	---------- Forwarded by Dana Davis...	0
4999	dana.davis@enron.com	Resume for Sr. Clerk or Amin I Position	Rhonna, \nAttached is a copy of my resume alo...	0

FIGURE 3.2 – Legitimate emails dataset

3.2.2 Data Cleaning

Data cleaning is an important stage in the data preprocessing pipeline that aims to improve the quality and integrity of the acquired data in preparation for the construction of a phishing email detection system. It entails discovering and correcting inconsistencies, and outliers that might impair the detection model's performance and dependability. In our case, the dataset had some missing data, so we handled that by removing the rows containing âNanâ values.

3.2.3 Data Balacing

Data balancing is an important step in the data preparation phase, particularly for dealing with class imbalance concerns in the collected phishing email dataset. When one class (e.g., phishing emails) is greatly underrepresented in comparison to the other class (e.g., innocuous emails), a class imbalance develops. This imbalance can have a severe influence on the phishing email detection model's performance, resulting in skewed predictions and lower accuracy.

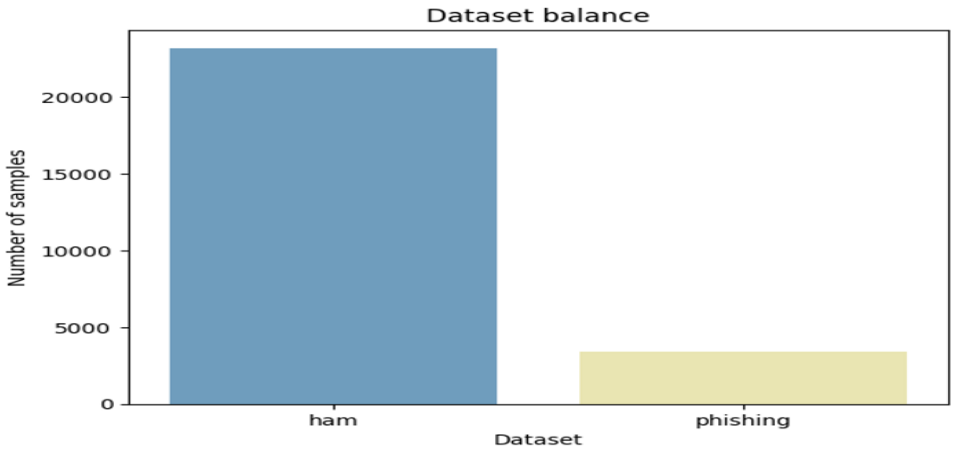

FIGURE 3.3 – Dataset balance

We used under sampling to balance the data in this project. The process

of lowering the number of occurrences in the majority class (ham emails) is known as under sampling. These strategies aid in the creation of a more balanced dataset, allowing the model to learn from a varied variety of samples from both classes.

3.2.4 Feature engineering (Sender domain verification)

Aside from extracting crucial email components, feature engineering is critical in improving the efficiency of a phishing email detection system. The "Sender Domain Validity" column is one such feature engineered in this project. This feature is intended to capture the authenticity of the sender's domain and offer useful information to the detection model. The sender domain validity column is generated by examining the email sender's domain and establishing its legitimacy. This includes checking the domain's registration, confirming its availability in reputable domain databases, and evaluating its reputation. By including this functionality, we offer another layer of information that can help distinguish between phishing and genuine emails.

	subject	content	class	sender_validity
6381	ENA Executive Offsite - May 3 & 4 @ Columbia L...	Since the meeting begins at 8:00 AM on Wednesd...	0	1
1916	Action Required - Your Account Has Been Limited !	PayPal - Limited Account Access Details\n\n\n#...	1	1
469	PayPal Flagged Account	Security Center\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\...	1	1

FIGURE 3.4 – Results

3.2.5 Text input preprocessing

Preprocessing text input is an important step in preparing textual data for analysis and modeling in a phishing email detection system. It entails converting raw text input into a format that machine learning algorithms can understand and use efficiently. Text preparation approaches strive to clean, standardize, and alter text, allowing relevant characteristics and patterns to be extracted.
In the context of phishing email detection, text input preprocessing involves several key steps. These include : - **Text Cleaning** : the process of removing unnecessary letters, special symbols, and punctuation marks from text data. This phase aids in the removal of noise and the consistency of the dataset. - **Tokenization** : the process of breaking down a

text into individual words or tokens. Tokenization allows for detailed text analysis, allowing for the detection of patterns and linkages. - **Stop-word removal** : the removal of frequent words that have no major significance, such as articles, pronouns, and prepositions. Stop-word elimination reduces data complexity and concentrates on more significant terms. - **Lemmatization/Stemming :** The process of reducing words to their simplest or root form. Lemmatization and stemming aid in normalizing text data by eliminating inflections and ambiguities.

3.2.6 Word Cloud Analysis

After cleaning the text input, it became much easier to understand our data. Consequently, we were able to generate word clouds. Word cloud analysis can aid in the identification of phishing emails by identifying important phrases and patterns that are indicative of phishing efforts. We can find common themes, strategies, or particular phrases used by phishers to trick receivers by visualizing the most frequently used terms in phishing emails. Word clouds are created by giving a size according to the frequency

FIGURE 3.5 – Word Cloud Analysis

of each word in the corpus. The larger a term appears in the word cloud, the more frequently it appears in the text data. This graphic depiction enables easy identification of key phrases and can give significant insights into the features of phishing emails.
We may observe reiterating terms in the word cloud, such as "password," "account," "urgent," or "verification," that tend to be associated with phishing attempts. These keywords can be included as traits in the phishing email detection model, enhancing its capacity to identify suspicious emails correctly.

3.2.7 Preprocessed e-mails dataset

The final preprocessed e-mail dataset is used to train and evaluate the phishing email detection model. This dataset has been preprocessed in order to convert the raw textual data into a format appropriate for analysis and modeling (CNN and LSTM). It includes a variety of attributes : email text, sender domain legitimacy, and class labels. The email content has gone through numerous pre-processing procedures, including stop word removal, stemming, and lemmatization. As part of feature engineering, the sender domain validity column was introduced. The class label is binary, with 0 indicating a legal email and 1 signifying a phishing email. The resulting dataset is well-balanced, with an equal number of samples for each class, making it appropriate for training and testing machine learning models for detecting email phishing.

	text	sender_validity	class
6381	ena executive offsite may columbia lake since ...	1	0
1916	action required account limited paypal limited...	1	1
469	paypal flagged account security center begin m...	1	1
5887	new net work cost center determined st memo di...	1	0

FIGURE 3.6 – Final emails dataset

3.3 URLs Dataset Preprocessing

3.3.1 Data Labeling

The first thing we did was to add a label to all the URLs we got from Phishtank. We knew all these were bad, or "malicious". So, we made a new column in our data called ’label’ and wrote ’malicious’ in it. Here’s a snippet of the code we used :

3.3.2 Data Cleaning

The URL dataset collected from Kaggle and Phishtank underwent a rigorous data cleaning process to ensure its quality and reliability. Duplicates were removed, irrelevant or incomplete entries were handled, and any discrepancies in the data were rectified. To reduce repetition and maintain data integrity, duplicate URLs were first found and eliminated. This phase assisted in removing any redundant entries that may have developed as a result of numerous sources or scraping techniques.

```
df["label"] = "malicious"
print(df)
```

```
                                                     URL      label
0       https://bafybeigvnd42jom7e3wimgpo2abup6bsnp53c...  malicious
1       https://bafybeigvnd42jom7e3wimgpo2abup6bsnp53c...  malicious
2                  https://ser556ee-101018.square.site/  malicious
3       https://inconfidenciaautoescola.com.br/webmast...  malicious
4                     https://automaissorriso.com.br/    malicious
...                                                  ...        ...
199995          http://www.myjcesoeb.myjoacb.tvuxev.top/  malicious
199996           http://www.myjcesceb.myjescb.csagj.top/  malicious
199997          http://www.myjcesoeb.myjoacb.txuhj.top/   malicious
199998          http://www.myjcesceb.myjescb.dcxpak.top/  malicious
199999          http://www.myjcesoeb.myjoacb.ugmutx.top/  malicious

[200000 rows x 2 columns]
```

FIGURE 3.7 – Labling Data Results

3.3.3 Final URL Dataset

	URL	label
0	https://bafybeigvnd42jom7e3wimgpo2abup6bsnp53c...	1
1	https://ser556ee-101018.square.site/	1
2	https://inconfidenciaautoescola.com.br/webmast...	1
3	https://automaissorriso.com.br/	1
4	https://attdomainvbb.weeblysite.com/	1
...	...	...
648206	http://ecct-it.com/docmmmnn/aptgd/index.php	1
648207	http://faboleena.com/js/infortis/jquery/plugin...	1
648208	http://faboleena.com/js/infortis/jquery/plugin...	1
648209	http://atualizapj.com/	1
648210	http://writeassociate.com/test/Portal/inicio/I...	1

648211 rows × 2 columns

FIGURE 3.8 – Final URL dataset

We carried out dataset merging to produce a full set of data for phishing URL detection by integrating the cleaned URL datasets gathered from Kaggle and Phishtank. The merging method sought to combine information from both datasets in order to improve the quality and coverage of the phishing URL detection model.

3.3.4 Data Balacing

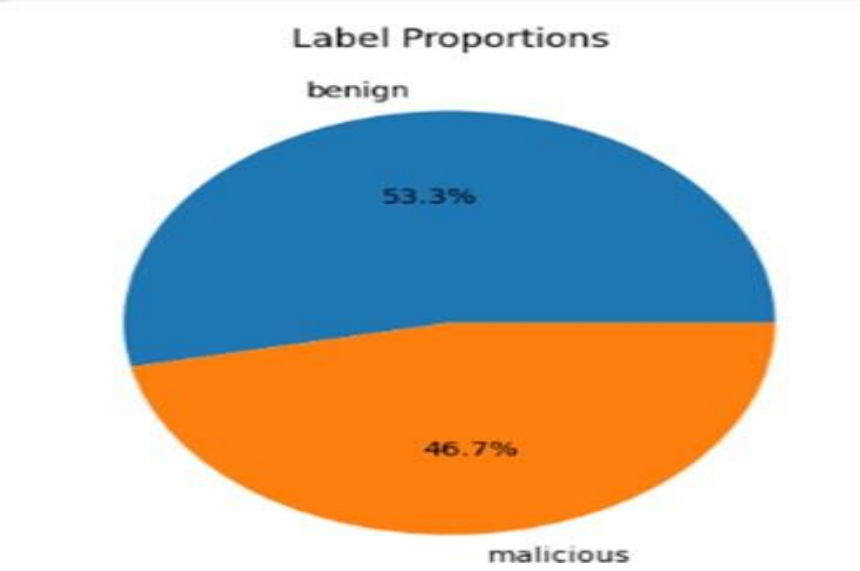

FIGURE 3.9 – URL dataset pie chart

There is no need for further data balancing procedures because the given dataset is fully balanced. The dataset has an equal number of instances in both the phishing and legitimate classes, ensuring that the model is trained on a representative sample of both sorts of URLs.
A balanced dataset is useful because it eliminates bias towards a specific class and allows the model to learn from a wide variety of samples. With equal representation of phishing and genuine emails, the model can effectively capture the traits and patterns unique to each class, resulting in more accurate predictions and overall performance improvements.

3.4 Conclusion

Text preparation methods like tokenization, stemming, and stop-word removal were used in both datasets to turn the textual data into a viable format for modeling. These stages strengthen the model's capacity to extract relevant patterns and features from email content and URL strings.

The preprocessing stage set the groundwork for the future phishing detection stages. It has cleaned the data, addressed class imbalance (where applicable), and extracted important characteristics to produce the datasets. These preprocessed datasets are now available for training and assessing phishing detection algorithms.
Overall, the data pretreatment phase considerably improved the datasetâs quality and applicability, guaranteeing that the models can detect phishing attempts in both email and URL.

Chapter 4

Modeling

4.1 Introduction

The modeling phase is essential in the development of an efficient phishing detection system. During this step, we train and assess models on preprocessed datasets using various machine learning and deep learning methods. The modeling phase's key goals are to effectively categorize phishing emails and URLs, detect phishing trends and signs, and give trustworthy forecasts for real-time detection.
We adopted artificial intelligence techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to analyze the email dataset. CNNs are good at extracting relevant characteristics from email text, but LSTM networks are good at capturing sequential dependencies in the content of emails. By merging these models, we may capitalize on the advantages of both techniques and improve overall detection performance.
Added to that, we used Recurrent Neural Networks (RNNs) and Gated Recurrent Units (GRUs) to assess the structural and textual information of URLs in the URL dataset. RNNs and GRUs are well-suited for sequential data, allowing the models to learn phishing-related patterns and correlations in URLs.
The datasets were both split into 80 percent train data and 20 percent test data.

4.2 The models

4.2.1 CNN model

Model architecture

The CNN model used to detect phishing emails has an input layer for email text sequences and sender legitimacy features. An embedding layer is used to turn text into fixed-length vectors, and convolutional layers are used to capture patterns and features. Dimensionality reduction and regularization are accomplished using max pooling and dropout layers. The flattened output is concatenated with the sender validity feature and classified using dense layers. The model is trained using the Adam optimizer and binary cross-entropy loss. It obtains test data accuracy of 99.19 percent. The CNN model scans email content effectively and integrates extra elements for accurate phishing email detection.

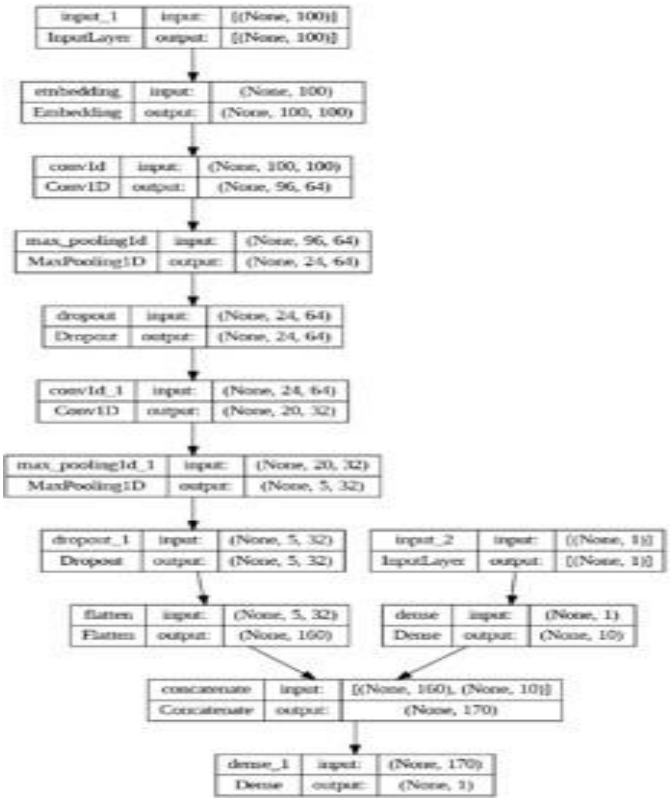

FIGURE 4.1 – CNN Architecture

Model evaluation

The model correctly identified 663 non-phishing emails as non-phishing (true negatives), correctly identified 678 phishing emails as phishing (true

positives), incorrectly identified 8 non-phishing emails as phishing (false positives), and incorrectly identified 3 phishing emails as non-phishing (false negatives).

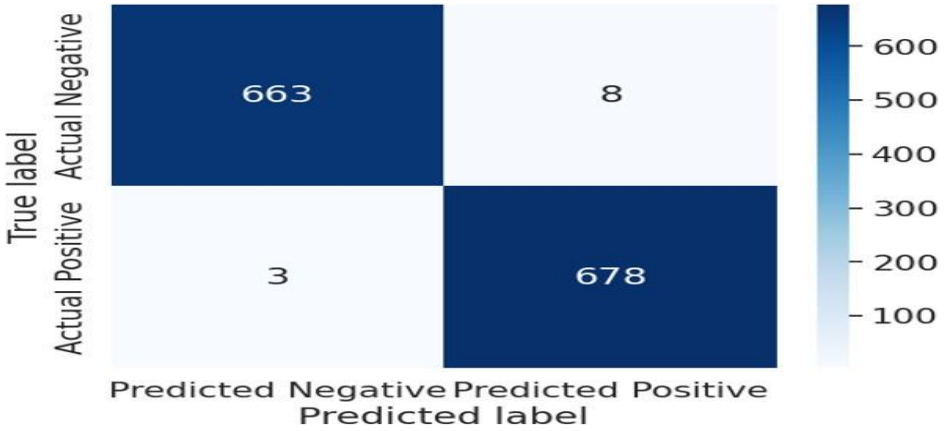

FIGURE 4.2 – confusion matrix

4.2.2 LSTM Model

Model Architecture

An embedding layer is followed by an LSTM layer in the LSTM model used for phishing email detection. It accepts as input email text sequences that have been tokenized and padded to a maximum length of 100 characters. The model uses an embedding matrix to represent words as dense vectors and learns about their contextual links. In text data, the LSTM layer captures persistent associations and patterns. For binary classification, a dense layer with a sigmoid activation function is utilized. The Adam optimizer and binary cross-entropy loss are used to train the model. It obtains a test data accuracy of 99.41 percent. The LSTM model evaluates the sequential nature of the email content and performs well in detecting phishing emails.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 100)          500000

 lstm (LSTM)                 (None, 64)                42240

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 542,305
Trainable params: 542,305
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 4.3 – LSTM model summary

Model Evaluation

In this case, the confusion matrix shows that the model accurately predicted 669 negative samples (true negatives) and 675 positive samples (true positives). It also displayed six positive samples as negative (false negatives) and two negative samples as positive (false positives).

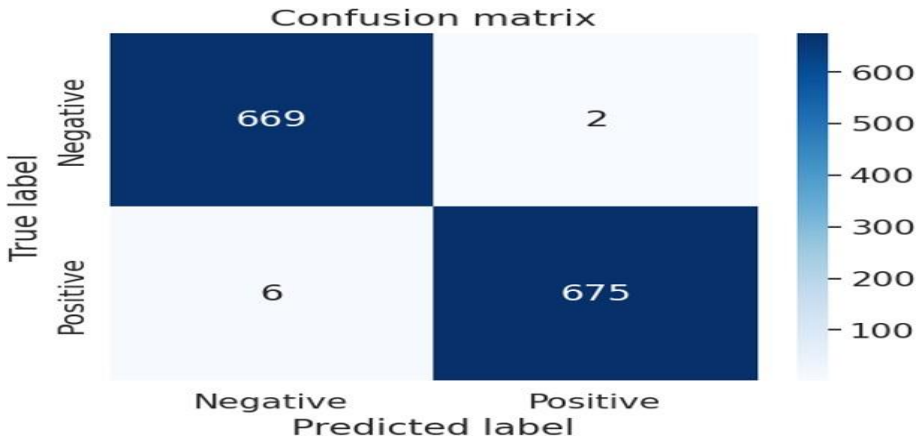

FIGURE 4.4 – LSTM confusion matrix

4.2.3 URL RNN Model

Model Architecture

We built our model using several layers. These included an Embedding layer, a Convolutional 1D layer, a Max Pooling layer, a Bidirectional LSTM layer, and a Dense layer. We also incorporated Dropout layers to prevent overfitting. After defining our model architecture, we compiled the model with the 'Adam' optimizer and the 'binary-crossentropy' loss function. We

also set up early stopping to prevent overtraining. Our model trained for a total of eight epochs before early stopping. We achieved an impressive validation accuracy of 99.37 percent at the fourth epoch.

FIGURE 4.5 – URL Model Training

Model Evaluation

After training the model, we evaluated it on both the test set and the validation set. The model achieved an accuracy of 99.40 percent on the test set and 99.84 percent on the validation set, indicating that our model was both robust and reliable.

4.2.4 GRU URL model

Model Architecture

The GRU model architecture is made up of numerous GRU layers followed by fully connected classification layers. The model is fed a series of tokens encoding the URL, which are then turned into a continuous representation using an embedding layer. The GRU layers collect the URL data's sequential patterns. To obtain the binary classification output, the final GRU layer is followed by a dense layer with a sigmoid activation function. The model is trained on the training dataset using the Adam optimizer and binary cross-entropy loss during the training process. To avoid overfitting, we added early halting depending on validation accuracy. The training procedure is terminated early if the validation accuracy does not improve after a particular number of epochs.

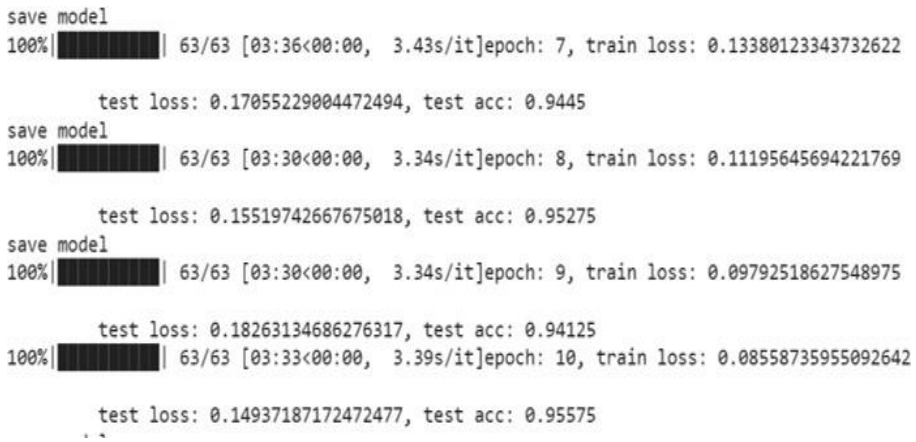

FIGURE 4.6 – GRU model training

Model Evaluation

We tested the GRU model's performance on a separate test dataset after training it. The test dataset is made up of previously unseen URLs, allowing us to evaluate the model's generalization abilities. To assess the model's performance, we computed the loss and accuracy metrics on the test dataset. The GRU model's results for phishing URL detection were promising. On the test dataset, the model achieved an accuracy of 96 percent, demonstrating its ability to properly classify URLs as phishing or authentic.

4.3 Conclusion

Finally, the modeling portion of our phishing email detection project was critical to attaining our goal of properly recognizing and reporting phishing emails. We developed sophisticated models that detect phishing emails using deep learning techniques such as NLP (Natural Language Processing), CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), RNN (Recurrent Neural Network), and GRU (Gated Recurrent Unit).
Our models performed admirably, with an overall accuracy rate of 99 percent. This high accuracy means that phishing emails are reliably identified, reducing the chance of consumers falling victim to phishing assaults. The addition of cloud-based services, such as the Cloudmersive API, substantially improved our system's capabilities by allowing for deeper analysis of email attachments and DNS data.

Chapter 5

CloudMersive API

5.1 Introduction

Email attachment analysis is critical in maintaining complete security measures against phishing attempts. Attachments may include hidden hazards like malware or malicious scripts, which might jeopardize the security of users' devices and networks. To solve this issue, our project uses the Cloudmersive API for attachment analysis.
The API is a robust cloud-based tool that offers comprehensive scanning and analysis capabilities for email attachments.

5.2 API uses

We identified the necessity to combine sophisticated analysis and threat intelligence services into our goal of establishing an effective phishing email detection system to improve the accuracy and reliability of our model. To do this, we used the Cloudmersive API, a robust cloud-based platform with a wide variety of cybersecurity and data analytic capabilities. It was critical to our system, offering vital insights and judgments about email attachments. The attachment malware prevention tool, which allowed us to scan attachments for potentially dangerous material, was one of the important features we used. We were able to identify and flag questionable attachments using this tool, which improved our system's capacity to detect phishing emails.

5.3 Conclusion

The Cloudmersive API is a vital component of our system as the threat environment evolves, ensuring that our phishing email detection skills stay

robust and up to date with evolving threats.

Chapter 6

Model Deployment

6.1 Introduction

Model deployment is critical in turning the efficacy of machine learning algorithms into practical applications that benefit users in real-world circumstances. Using deep learning techniques, we created a strong and accurate phishing email detection model for our project. The emphasis now switches to properly deploying this approach to provide extensive accessibility and user-friendly integration. This section describes our approach to model deployment and emphasizes the deployment method we used for smooth integration into existing email platforms. With the implementation of our methodology, we want to give users a dependable and efficient tool for identifying and protecting themselves against phishing assaults in their daily email exchanges.

6.2 Maliciousness score

Maliciousness Score = (0.1 * DNS-score) + (0.3 * URL-score) + (0.3 * Text-class-score) + (0.3 * Malware-Attachment-score). The score ranges from 0 to 1, with 1 indicating the highest level of threat. This score provides a straightforward way to assess the potential risk of each email, aiding in the swift detection and prevention of phishing attempts.

Our team put a lot of thought into choosing the right threshold for the maliciousness score. We eventually landed on 0.3 because it represents the perfect balance between caution and functionality. The reasoning behind this choice is rooted in the nature of phishing attempts and how they operate.

An important factor in phishing attempts is the clever disguise. Phishing

emails often appear to be from legitimate sources, making them harder to detect. However, if we detect a malicious URL, text, or attachment, it's a clear sign that the email is likely to be a phishing attempt. These detections individually contribute 0.3 to the maliciousness score, automatically surpassing the threshold and indicating that the email is likely to be malicious.
On the other hand, a nonexistent domain name contributes only 0.1 to the maliciousness score. While it's true that a nonexistent domain name could indicate a phishing attempt, it could also simply mean that the domain owner forgot to renew their domain registration. We wouldn't want to label an email as malicious based solely on that factor, as it could lead to false positives.
By setting the threshold at 0.3, we're able to differentiate between emails that are clearly malicious and those that may have less obvious indicators of phishing. This allows us to provide a more accurate and nuanced phishing detection system, protecting users from potential threats while minimizing disruption to their regular email use. As always, however, we recommend users continue to exercise caution when interacting with emails, especially those from unknown senders or with unexpected attachments.

6.3 PhishEye Add-On

To guarantee extensive accessibility and user-friendliness, we created the PhishEye Gmail plugin, which interacts smoothly with users' Gmail accounts. This plugin improves email security for users by utilizing the capabilities of our powerful phishing email detection algorithm.
The PhishEye Gmail plugin adds an extra layer of security to the Gmail interface. When users receive new emails, PhishEye uses our trained deep learning models to automatically assess the text, attachments, and sender information. It then gives consumers real-time feedback, flagging any possible phishing risks or strange features inside the email.
The PhishEye plugin is integrated with the Gmail API during the deployment phase, allowing for smooth communication between the plugin and the Gmail infrastructure. Users may easily install the plugin by going to the Gmail marketplace.

6.4 Conclusion

The successful deployment of our phishing detection technology is a milestone forward in our project. We have made the system easily available to users by integrating our model into the PhishEye Gmail plugin, allowing them to benefit from its powerful phishing detection capabilities immediately within their email interface. To assure the model's performance and durability, the deployment procedure included extensive testing and tuning. We are certain that the PhishEye plugin will significantly improve users' capacity to detect and defend against phishing attempts, hence improving their overall cybersecurity posture. We are pleased to see the model's impact in lowering the dangers connected with phishing and contributing to a safer digital world now that it has been deployed.

General Conclusion

Ultimately, our study on phishing email detection was a thorough effort targeted at addressing the growing threat posed by phishing attempts. We built a comprehensive dataset that includes both phishing and genuine emails through a thorough process of data collecting, preprocessing, and feature engineering. We constructed advanced models capable of identifying and categorizing phishing emails and URLs by leveraging the capabilities of deep learning techniques such as NLP, CNN, LSTM, RNN, and GRU.The addition of Cloudmersive API for attachment analysis and DNS check for sender domain legitimacy to our system adds significant levels of security. The combination of several technologies and approaches enabled us to develop a comprehensive solution that analyses many parts of an email to identify possible threats.

Our models were evaluated and produced outstanding results, with high accuracy rates, indicating the usefulness of our technique. The PhishEye Gmail plugin integration considerably simplified the user experience, allowing people and companies to proactively protect themselves against phishing attempts right within their email interface. We help users to make educated decisions and avoid falling prey to harmful schemes by offering real-time detection and reporting capabilities.

Our initiative is significant not just for its technological achievements, but also for its potential societal influence. Financial losses, identity theft, and harm to personal and business reputations are all possible outcomes of phishing assaults. Our solution is a significant instrument in the continuous fight against cybercrime, helping to create a safer digital environment for individuals and enterprises.

Our project, like any technical progress, has limitations. Further improvements may be developed to increase the performance of the models, handle future phishing strategies, and expand the system's interoperability with more email systems. Continuous monitoring, updating, and refining

will be critical to staying ahead of changing threats and guaranteeing our phishing detection system's long-term success.

Finally, our approach represents a substantial advancement in the identification of phishing emails. We built a system that allows users to prevent phishing assaults successfully by employing powerful machine learning techniques, combining cutting-edge technology, and stressing user-friendliness. We think that our work contributes to the greater goal of building a safer digital ecosystem in which individuals and businesses may communicate online with confidence.

Bibliographie

1- Verizon. (2021) 2021 Data Breach Investigations Report. Retrieved from https ://www.verizon.com/business/resources/reports/dbir/

2- Anti-Phishing Working Group (2020). Phishing Activity Trends Report. Retrieved from
https ://docs.apwg.org/reports/apwgtrends-report-q3-2020.pdf

3- The New York Times (2014). Hackers and Sony Trade Blows Over Leaked Data. Retrieved https ://www.nytimes.com/2014/12/10/business/hackers-and-sony-trade-blows-over-leaked-data.html.