A Predictive Model for Phishing Detection Based on Convolutional Neural Networks

by

Keltoum Hamaidi, B.Sc.

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science (M.Sc.)

to the

Department of Computer Science
Computer & Networks Security Specialization
Faculty of Engineering Sciences
University of Badji Mokhtar – Annaba

Supervised by: **MCA Mohamed Hafidi**
Defense Chair: **MCB Nafa Mehdi**
Examiner:**MCB Ahmim Marwa**

Annaba, 2021

"You can fool some of the people all of the time, and all of the people some of the time, but you cannot fool all of the people all of the time."

–Abraham Lincoln

Acknowledgement

First of all, my most sincere thanks go to “ALLAH” Almighty who gave me the great will and adequate knowledge to carry out this modest work.

My sincere thanks go to my supervisor Dr. Mohamed Hafidi, whose advice and guidance have been invaluable to me in carrying out this project.

I respectfully thank all the members of the jury.

Finally, my thanks go to all those who contributed directly or indirectly to the accomplishment of this work.

Dedication

I dedicate this modest work as a sign of respect, recognition and thanks

to my dear parents.

My mother who has worked for my success, through her love, her support, all the sacrifices she has made and her precious advice. For all your assistance and your presence in my life, receive through this work, however modest it may be, the expression of my feelings and my eternal gratitude.

My father who can be proud and find here the result of long years of sacrifices, that have helped me progress in my life. May God ensure that this work pay him back a little. Thank you for the noble values, education and permanent support from You.

To my dear brother & sisters.

To all my dear friends.

Table of Contents

CHAPTER 1: THE PHISHING

CHAPTER 2: MACHINE LEARNING

Table of figures

List of Tables

Abreviations

- **APWG:** Anti-Phishing Working Group
- **RSA:** Rivest, Shamir, & Adleman (public key encryption technology)
- **SAAS:** Software as a Service
- **URL:** Uniform Resource Locator
- **SMS:** Short Message Service
- **CEO:** Business email compromise
- **HR:** human resources
- **WWW:** World Wide Web
- **IP:** Internet Protocol
- **AI:** Artificial intelligence
- **CNN:** Convolutional neural network
- **SPN:** The sum–product network
- **ML:** Machine Learning
- **SVM:** Support Vector Machine
- **PDRCNN:** Precise Phishing Detection with Recurrent Convolutional Neural Networks
- **LSTM:** Long short-term memory
- **ccTLD:** country-code top-level domains
- **SLD:** second-level domain
- **ANN:** Artificial neural network
- **Gnuplot:** command-driven interactive function plotting program
- **GNB:** Generalized Naïve Bayes
- **DT:** Decision Tree
- **LR:** Logistic Regression
- **RF:** Random Forest
- **VM:** Virtual Machine

Abstract

Internet scams are many and varied. Anyone is likely to be the target of an attack while browsing the net. More and more scammers are quick to resort to social engineering as a lever to gain unfairly sensitive data by exploiting human weaknesses.

Phishing is a Social Engineering technique employed by these hackers. It is used to steal personal information to commit identity theft without the knowledge of its victims. The persuasiveness of these scammers is the keystone of a successful attack.

In this time of global crisis, phishing attacks have boosted by the increase use of technology in every sector. The continuous growth and the rising volume of phishing websites have led to individuals and organizations worldwide becoming more and more exposed to various cyberattacks. Consequently, more effective phishing detection is required for improved cyber defense. In this project, I will be presenting a machine learning-based approach to enable high accuracy detection of phishing sites by combining several features to attain the best results. The proposed method utilizes **Convolutional Neural Networks** (**CNN**s) for high accuracy classification to distinguish legitimate sites from phishing sites. We evaluate the models using a dataset obtained from 11055 datapoints with 6157 legitimate websites and 4898 phishing websites. Based on extensive experiments, my **CNN**-based model proved to be highly effective in detecting unknown phishing sites with a 97,06% accuracy. I have also built a HTML5 and Bootstrap CSS frontend and deployed the backend of my user-friendly web platform by using a Flask Framework.

Keywords—**Phishing; Convolutional Neural Networks; Machine learning; Deep learning; Phishing website detection; Social Engineering**

Résumé

Les actes frauduleux sur Internet sont nombreux et variées. N'importe qui est susceptible d'être la cible d'une attaque en naviguant sur le web. De plus en plus hameçonneurs recourent rapidement à l'ingénierie sociale comme levier pour obtenir des données injustement sensibles en exploitant les faiblesses humaines.

Le phishing est une technique d'ingénierie sociale employée par ces pirates. Il est utilisé pour voler des informations personnelles afin de commettre une usurpation d'identité à l'insu de ses victimes. La force de persuasion de ces escrocs est la clé de voûte d'une attaque réussie.

En cette période de crise mondiale, les attaques de phishing ont été stimulées par l'utilisation croissante de la technologie dans tous les secteurs. La croissance continue et le volume croissant des sites Web de phishing ont conduit les individus et les organisations du monde entier à être de plus en plus exposés à diverses cyberattaques. Par conséquent, une détection plus efficace du phishing est nécessaire pour une meilleure cyberdéfense. Dans ce projet, je présenterai une approche basée sur l'apprentissage automatique pour permettre une détection de haute précision des sites de phishing en combinant plusieurs fonctionnalités pour obtenir les meilleurs résultats. La méthode proposée utilise des réseaux de neurones convolutifs (**CNN**) pour une classification de haute précision afin de distinguer les sites légitimes des sites de phishing. Nous évaluons les modèles à l'aide d'un ensemble de données obtenu à partir de 11055 points de données avec 6157 sites Web légitimes et 4898 sites Web de phishing. Sur la base d'expériences approfondies, notre modèle basé sur **CNN** s'est avéré très efficace avec plus que 97% de précision pour détecter les sites de phishing inconnus. J'ai également construit un frontend HTML5 et Bootstrap CSS et déployé le backend de ma plate-forme Web en utilisant un Flask Framework.

Mots-clés — hameçonnage ; Réseaux de neurones convolutifs ; Apprentissage automatique ; L'apprentissage en profondeur ; Détection de sites Web d'hameçonnage ; Ingénierie sociale

المقدمة

عمليات الاحتيال عبر الإنترنت كثيرة ومتنوعة. من المحتمل أن يكون أي شخص هدفًا لهجوم أثناء تصفح الإنترنت. يسارع الكثير من المحتالين إلى اللجوء إلى الهندسة الاجتماعية كأداة للحصول على بيانات حساسة دون وجه حق من خلال استغلال نقاط الضعف البشرية. تصيد المعلومات الاحتيالي هو أحد تقنيات الهندسة الاجتماعية التي يستخدمها هؤلاء المخترقين. يتم استخدامه لسرقة المعلومات الشخصية لارتكاب سرقات الهويات دون علم ضحاياه. مهارة الإقناع عند هؤلاء هي حجر الأساس لأي هجوم ناجح. في هذه الفترة من الأزمة العالمية، تعزز وجود هجمات من هذا النوع من خلال زيادة استخدام التكنولوجيا في كل قطاع. أدى النمو المستمر والحجم المتزايد لمواقع التصيد الاحتيالي إلى زيادة تعرض الأفراد والمؤسسات في جميع أنحاء العالم للهجمات الإلكترونية المختلفة. وبالتالي، فإن الكشف الفعال لمحاولات التصيد الاحتيالي مطلوب لتحسين الدفاع الإلكتروني. في هذا المشروع. سأقدم نهجًا قائمًا على التعلم الآلي لتمكين الكشف عن مواقع التصيد بدقة عالية من خلال الجمع بين العديد من الميزات لتحقيق أفضل النتائج. تستخدم الطريقة المقترحة الشبكات العصبية التلافيفية, لتصنيف الدقة العالية لتمييز المواقع الرسمية عن مواقع التصيد الاحتيالي. نقوم بتقييم النماذج باستخدام مجموعة بيانات تم الحصول عليها من 11055 نقطة بيانات مع 6157 موقع ويب رسمي و4898 موقعًا للتصيد الاحتيالي. استنادًا إلى تجارب مكثفة، أثبت نموذجنا المستند على فعاليته العالية في اكتشاف مواقع التصيد غير المعروفة.

الكلمات الرئيسية ـ التصيد الاحتيالي. الشبكات العصبية التلافيفية التعلم الالي؛ تعلم عميق؛ الكشف عن مواقع التصيد الاحتيالي؛ هندسة اجتماعية

.

General Introduction

The pandemic COVID-19 has boosted the use of technology in every sector. Phishing is a fraudulent attempt to obtain sensitive information such as usernames, passwords, and bank account details, often for malicious reasons. It is usually done through email spoofing or instant messaging, and it often tricks users into entering personal information on a bogus website, which looks and feels the same as the legitimate site, the only difference being the URL of the relevant website. Communications purported to come from social websites, auction sites, banks, and online payment processors are often used to lure victims. Phishing emails can contain links to websites that distribute malware.

Detection of phishing websites often includes searching a directory of malicious sites. Since most phishing websites are short-lived, the directory cannot always keep track of all of them, including new phishing websites. Thus, the problem of detecting phishing websites can be better solved by **machine learning** techniques. Based on a comparison of different ML techniques, the convolutional neural network seems to work better.

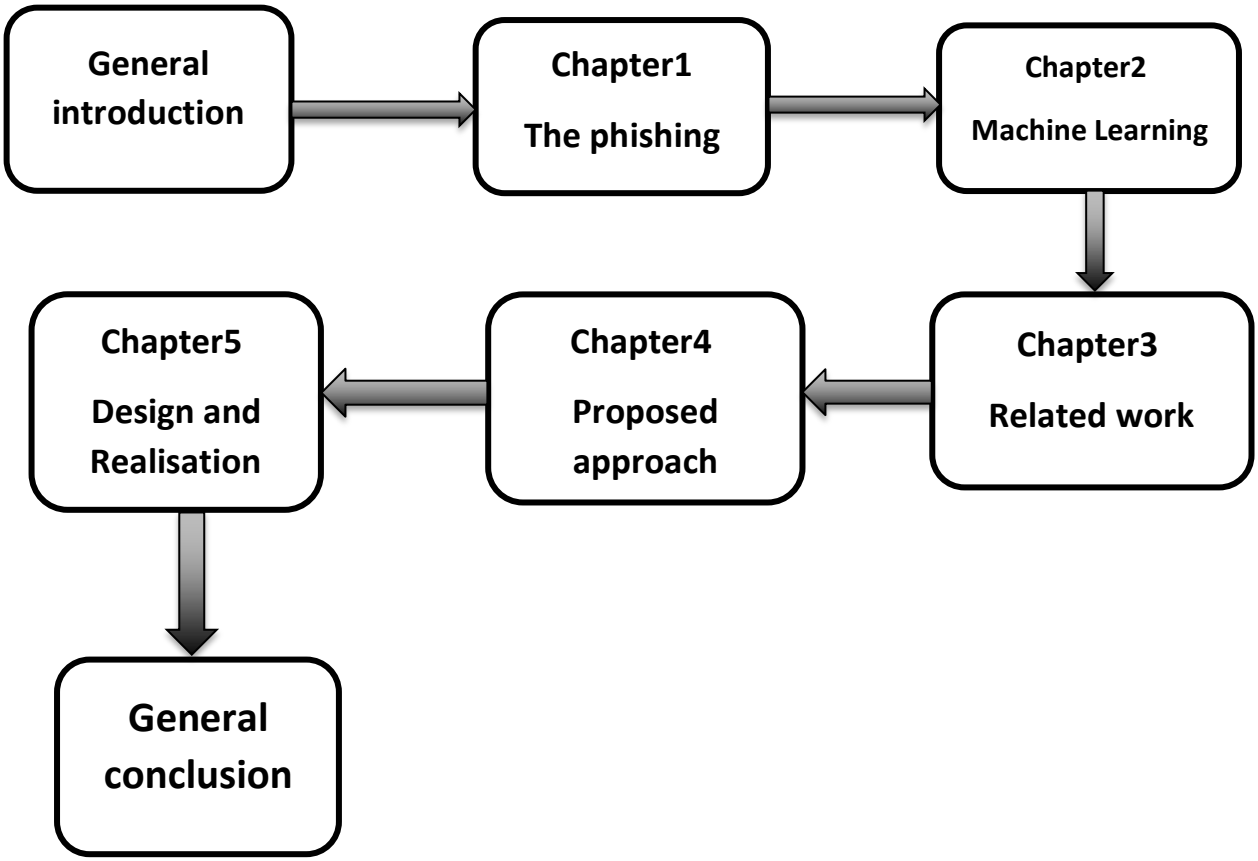
- Phishing is defined as "the fraudulent act of acquiring private and sensitive data, masquerading as a trustworthy entity. Generally, the target data are of the following types: login credentials, passwords, credit cards, or social security numbers. "The choice of this issue is justified by several reasons:

-This type of attack (phishing) is the most commonly used on the Internet for identity theft.

-Despite the existing tools, the number of victims continues to grow.

-The direct and horrendous effects phishing causes on victims.

This thesis consists of five chapters that are organized as follows:


General introduction
Chapter1
The phishing
Chapter2
Machine Learning
Chapter3
Related work
Chapter4
Proposed approach
Chapter5
Design and Realisation
General conclusion

Chapter 1

The Phishing

1. Introduction

Phishing is defined as a cyber-attack form that is often used to steal user data.

Typically, attackers use phishing emails to distribute crafted links or attachments, which they can then use to perform a variety of functions. For example, criminals try to get access to victims' data or account information. These attempts are made to impersonate a trustworthy communication partner in electronic communication via fake websites, e-mails or short messages.

Lately phishing attacks have gained a lot of energy all around the world. The Anti-Phishing Working Group (APWG) has reported many unique phishing websites. Alone in the first quarter of 2018, the number of phishing detections has increased by 46% compared to the fourth quarter of 2017 [1]. Another report by RSA has estimated that global organizations have suffered losses amounting to $9 billion due to phishing incidents in 2016 [2]. Lately in this Global quarantine [3], there has been a record in terms of social engineered attacks. These attacks have caused the misfortune of millions of dollars at a worldwide level. For this reason, it is essential to distinguish these cyber risks and take appropriate measures against them in order to keep individuals and corporate information secure. These statistics say much about the ineffectiveness, to a certain level, of the existent anti-phishing solutions. With the increasing era of digitization, there is an urge to boost the anti-phishing solutions to provide a safe cyberspace experience.

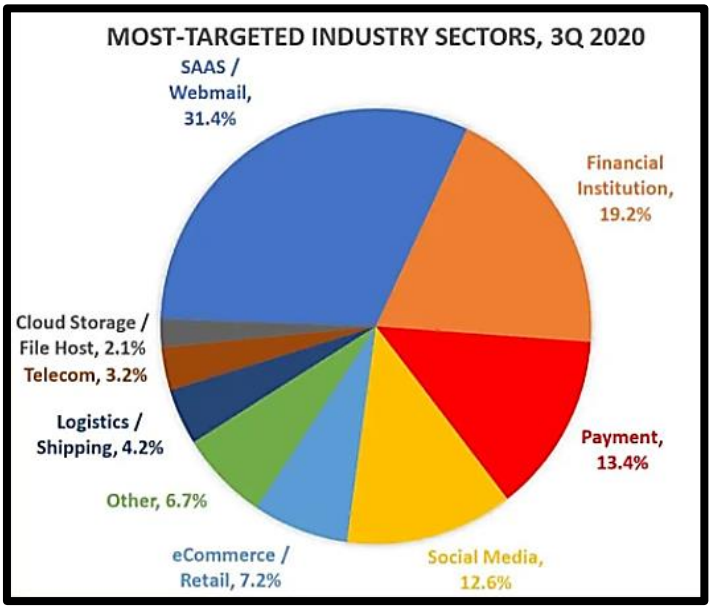


Figure 1: Phishing activity trends report 3rd

As shown in Fig.1, the report from PhishLabs the most widely used and influential mean is the phishing email. Phishing mail is about using a fake email from a phisher to trick the user into returning information such as an account password. The attacker may send a particular fake web page such as bank web page that looks the same to convince users to enter their personal sensitive information like credit card or bank account number. phishing emails harm's is immense. In the United States alone, phishing emails are expected to bring a loss of 500 million dollars per year [25]. According to the APWG, the number of phishing emails increased from 68,270 in 2014 to 106,421 in 2015, and the number of different phishing emails reported from January to June 2017 was approximately 100,000 [26], [27]. In addition, Gartner's report notes that the number of users who have ever received phishing emails has reached a total of 109 billion [28]. Microsoft analyzes and scans over 470 billion emails in Office 365 every month to find phishing and malware. From January to December 2018, the proportion of inbound emails that were phishing emails increased by 250% [29].

2. Cyber security

Cybersecurity is the entirety of all technologies, processes, and procedures intended to protect networks, computers, programs, and data from attacks, damage, or unauthorized access.

The goal of implementing cybersecurity is to ensure a good security status for computers, servers, networks, mobile devices and the data stored on these devices. This includes protection against attackers with less honest intentions. Corresponding cyber attacks can be aimed at stealing the sensitive data of a company or a user. Use it to misuse it for other purposes or to blackmail companies and users alike. Therefore, cybersecurity is a critical factor for government organizations such as private companies and especially critical areas such as healthcare or finance.[33]

3. The cybercrime

The United Nations defines cybercrime as: "Any illegal behavior involving electronic operations aimed at the security of computer systems and the data they process" [4].

Cybercrime is defined as a crime in which a computer is the object of the crime (hacking, phishing, spam) or it is used as a mean of committing a crime. Rose Colin states in an article "Essay in the Notion of Cybercrime" that "Cybercrime is the third greatest threat to great powers, after chemical, bacteriological and nuclear weapons" [5]. Cybercrime is generally divided into two categories:

- Crimes targeting computer networks or devices. These types of crimes include viruses and denial of service scams.
- Crimes that use computer networks to promote other criminal activity. These types of crimes include cyber-harassment, phishing, and fraud or identity theft [6].

4. A spam

Spam is unsolicited bulk e-mail that is distributed on the Internet. The person who engages in this practice is a spammer. These spam e-mails are sent to millions of e-mail addresses. Most spam emails have a commercial background and are divided into the following types:

-commercial email spam.

-Chain letters / virus warnings / hoax.

-Emails sent by viruses.

-Phishing emails.

5. The Phishing

Phishing is the attempt to fraudulently obtain personal information or other private information from another person. Phishing is probably the most common form of internet fraud. Usually these are fraudulent emails or websites designed to trick potential victims into sharing their sensitive information with the scammer. Instead of using the stolen information for themselves, many scammers sell it on the dark net, mainly to hackers and cyber criminals who specialize in identity theft. With advances in cyber security, many cyber threats have come and gone, but phishing remains strong. The main reason phishing attacks are so common is because of the use of counterfeiting, tampering, and social engineering methods to deceive potential victims. Typically, phishing emails are written as urgent (albeit fake) notifications from internet service providers, digital wallets, financial institutions, and other organizations. In addition, many of them contain logos and other official images.

The scammers responsible for these attacks ask the potential victims to provide important information - be it social security number, credit card details or login details. To add a sense of urgency to their message, they provide an important reason why the victim should. For example, they could lose access to their bank account or be banned from one of their social media profiles if they use, they do not provide the requested information within the specified period.

To get the information they need, phishers create fake websites that look just like the real ones. They also have very similar URLs, which make it even harder for victims to spot the fakeness. According to the latest statistics, more than 1.5 million new phishing pages are created every month, with an average lifespan of three to five days per page. That is almost 50,000 new pages every day. So, it is no surprise that phishing is the number one culprit of data breaches around the world.

6. Phishing Techniques

There are several types of phishing scams, some of which can only be done by phone (i.e., voice phishing or vishing) or text messages (SMS phishing or SMiShing). The five most common types of online phishing scams include:

6.1 Spray-and-pray phishing

Commonly known as deceptive phishing, this is known as spray-and-pray. It is the oldest and most primitive form of online phishing. Phishers use this technique to send a series of "urgent" emails asking the potential victim to update their PayPal password or enter their details in order to win the lottery. These emails usually contain links to fake login pages. When a victim enters their personal information into these fake forms, it is instantly stored on a remote server that the phisher can access [8].

6.2 Spear phishing

Spear phishing is made far more complex and sophisticated because it is more personalized. Instead of sending a general message, phishers target specific organizations, groups, or even individuals with the aim of obtaining their personal information. They collect names, email addresses, and other personal information from network sites like LinkedIn or hacked email entries. This type of phishing is primarily aimed at businesses and organizations. Hence, spear phishing emails are different from deceptive emails. Although they have a similar layout, spear phishing emails usually contain false queries or invoices from business partners. Phishers claim that they have attached an important document and ask the victim to download it on their computer. When it does, malicious software is installed that spies on your activities and collects your personal information.

6.3 CEO Phishing

CEO phishing is a very sophisticated form of online fraud that can be very time consuming for the phisher behind it. These are cyber criminals who target people in HR or finance in an organization and act as either the company's CEO or another high-level executive. You exchange several messages with the victim to build trust.

After a while, the phisher suddenly asks his target to give him the personal information of the employees or, more often, to transfer money to an account specified by them. In most cases, they will say they need the funds on a new contract and claim the referral is very urgent. As outrageous as it sounds, companies around the world have lost up to $ 5 billion to phishing CEOs.

6.4 File Hosting Phishing

Many people use online hosting services like Dropbox and Google Drive to back up their files for easy access and sharing. Phishers understand why there have been countless attempts to compromise their victims' credentials. The layout of the fraud is essentially the same as the deceptive phishing, as it is a fake login page. However, instead of looking for something in particular, hackers want to access their victims' online file storage space in order to gather valuable information that they can find there.

6.5 Cryptocurrency Phishing

Cryptocurrency phishing is a relatively new form of online fraud. To set this in motion, hackers create fake login pages for cryptocurrency websites. When unsuspecting users enter their credentials through these fake sites, the hackers gain instant access to their victims' digital accounts and can withdraw funds in a matter of seconds. So far there has only been one major phishing attack on the cryptocurrency. However, as the digital currency increases, it can be assumed that there will be more of it in the future.

7. Website Spoofing

Web site spoofing allows an attacker to create a "shadow copy" of the entire WWW. Access to the shadow web is channeled through the attacker's machine, allowing it to monitor all victim activity, including passwords or account numbers entered by the victim. The attacker can also cause false or deceptive data to be sent to web servers on behalf of the victim, or to

the victim on behalf of any web server. In short, the attacker observes and controls everything the victim does on the web [9].

8. Characteristics of Phishing Domains

The figure below shows the structure of the URL, we will present it to understand how attackers proceed when they create a phishing domain.

It starts with a protocol used to access the page. The domain name identifies the server that is hosting the web page.

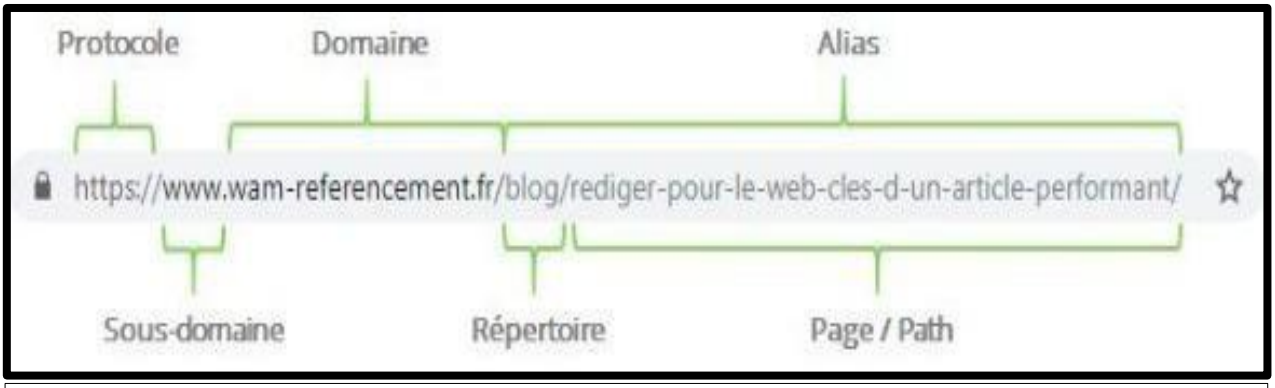


Figure 2: URL structure

- Sous-Domaine: The subdomain in a URL tells your browser which page of your site to display. For example, a subdomain such as "blog" provides the blog page for your website.
- The main domain name: it’s simply the name of the website. It is often similar to the brand name. People visiting "lvmh.com" will know directly that it is the LVMH brand, Louis Vuitton Moët Hennessy.

In addition, subdomains divide your website into different categories of content, and show Google and your visitors that it contains more information than just the home page.

A hacker controls the subdomain and the path and can change them. The latter two are completely controllable by the phisher. The phishers can register any domain name that has not been registered before.

After detecting phishing threats some companies specializing in computer security publish fraudulent web pages and IP addresses in the form of blacklists to warn web service users of the risks of these web sites.

9. Conclusion

In this chapter I have defined cybercrime and in particular phishing. I have cited the different types of phishing attacks. Finally, I have cited the parts of the URL that can be modified by the hacker to deceive users and the techniques to predict phishing websites.

Chapter 2
Machine Learning

1 Introduction

Machine Learning is a vast field based on concepts of computer science, statistics, cognitive sciences, engineering, optimization theory and many other disciplines. math and science. There are many applications for **machine learning**, but data mining is the most important of all [5]. It can mainly be classified into two broad categories: supervised **machine learning** and unsupervised **machine learning** [11].

2 What is Artificial Intelligence?

AI is the attempt to transfer human learning and thinking to the computer and thus to give it intelligence. Instead of being programmed for every purpose, an AI can find answers independently and solve problems independently.[57]

The aim of AI research has always been to understand the function of our brain and our mind on the one hand and to be able to reproduce them artificially on the other.

3 Definition

Machine Learning is the science and art of programming computers so that they can learn from data. Arthur Samuel the American pioneer of **machine learning** and artificial intelligence said in 1959: "**Machine learning** is the discipline that gives computers the ability to learn without being explicitly programmed" [12]. Added the American computer scientist Tom Mitchell in 1997 by a technical expression: “Given a task T and a performance measure P, we say that a computer program learns from an experiment E if the results obtained on T, measured by P, improve with experience E” [12]. The field of application of **machine learning** is very varied: the prediction of financial values, the detection of intrusions in the field of computer security, the detection of machine theft, the implementation of an anti-virus and the cryptanalysis [11]. There are three types of **machine learning**: supervised, unsupervised, and reinforcement learning.

4 Steps of Machine learning

4.1 Data Collection

is our first real step in **machine learning**, gathering the data. This step is very important because the quality and quantity of the data that you gather will directly determine how good your predictive model can be.

the data we collect will yield us a representation of data (table of content) and will be used for training.

4.2 Data Preparation

This is the step where we load our data into a suitable place and prepare it for use in our **machine learning** training.

Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions...)

put all our data together then randomize the order we wouldn't want the order of our data to affect how we learned and visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis and cut into training and evaluation sets.

4.3 Choose a Model

Different algorithms are for different tasks; choose the right one.

4.4 Train the Model

-This step is to answer a question and make an expectation correctly, we'll use their data to incrementally improve our model's ability to predict.

-Linear regression example: algorithm would need to learn values for W and b.

-Each iteration of process is a training step.

4.5 Evaluate the Model

Allows us to test our model against data that has never been used for training.

This metric allows us to see how the model might perform against data that has not yet seen, his is meant to be representative of how the model, might perform in the real world. A good rule that many researchers use for training-evaluation split is somewhere on the order 80%-20% or 70%-30%.

4.6 Parameter Tuning

If you want to see if you can further improve your training in any way you can do this by tuning some of our parameters (hyperparameter tuning), there were few that we implicitly when we did our training, and now it's a good time to go back and test those assumptions, try other values.

4.7 Make Predictions

Machine learning is using data to answer questions, prediction is that step where we finally get to answer some questions. This is the point, of all of this work where the value of **machine learning** is realized. We can finally use our model to make predictions.

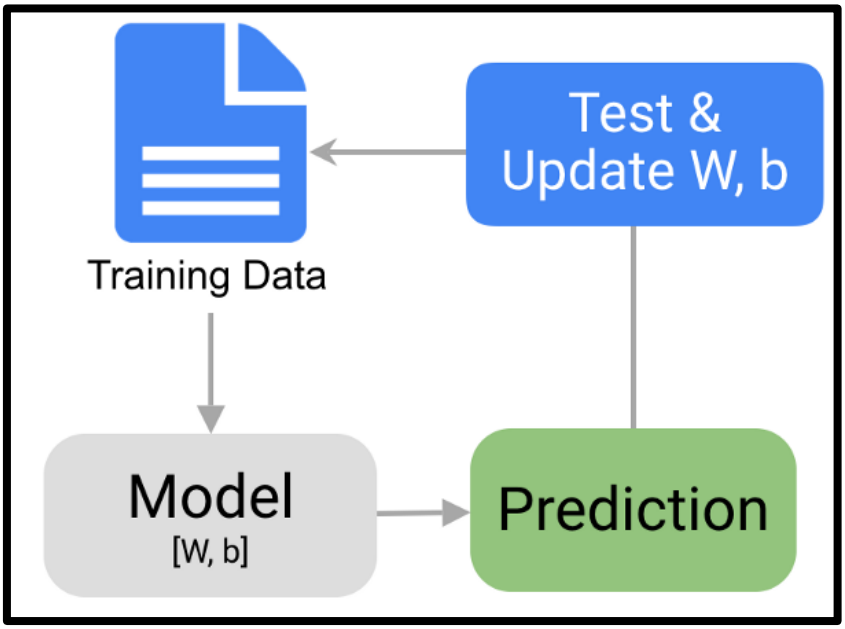


Figure 3: Steps of Machine learning

5 Types of machine learning

5.1 Supervised Learning

In the context of artificial intelligence (AI) and **machine learning**, supervised learning is a method in which both input and desired output data are provided. Input and output data are marked for classification in order to create a learning basis for future data processing. The term supervised learning comes from the idea that an algorithm learns from a training data set that can be thought of as a kind of teacher.

Input and output data are marked for classification in order to create a learning basis for future data processing. In most cases, the teacher (developer) provides the correct function value for an input. The algorithm learns a function from the pairs of input and output data. Monitored **machine learning** systems provide the learning algorithms with known data to support future decisions.

Chatbots, self-driving cars, facial recognition programs, expert systems, and robots are some of the systems that use either supervised or unsupervised learning. Supervised learning systems are mostly connected to retrieval AI systems, but can also use a generative learning model.

In general, supervised learning is used when a system is given input and output variables with the intent that it learns how these variables belong together. The goal is to create an accurate mapping function that allows the algorithm to predict the output when a new input is made.

This is an iterative process and every time the algorithm makes a prediction it is corrected or given feedback until it reaches an acceptable level of performance. The training data for supervised learning includes a number of examples with paired input topics and desired output (also known as the supervision signal). For example, an AI application for image processing that uses supervised learning can be equipped with labeled images of vehicles in categories such as cars or trucks. After the learning process, the system should be able to distinguish unlabeled images. The system should also recognize when the learning process is

complete. So, these networks are usually characterized by the subsequent allocations of classes conditioned on the visible data. The sum–product network (SPN) and Convolutional Neural Network (CNN) are examples of supervised deep network.

5.1.1 Classification techniques

"These techniques predict discrete variables. Classification models categorize input data. " [8] The classification method is used if the data can be marked or categorized into specific groups. There are four main types of classification tasks, they are:

- Binary Classification: **binary classification tasks involve one class that is the normal state and another class that is the abnormal state.**
- Multi-Class Classification: **Multi-class classification refers to those classification tasks that have more than two class labels.**
- Multi-Label Classification: **Multi-label classification refers to those classification tasks that have two or more class labels, where one or more class labels may be predicted for each example.**
- Imbalanced Classification: **Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed.**

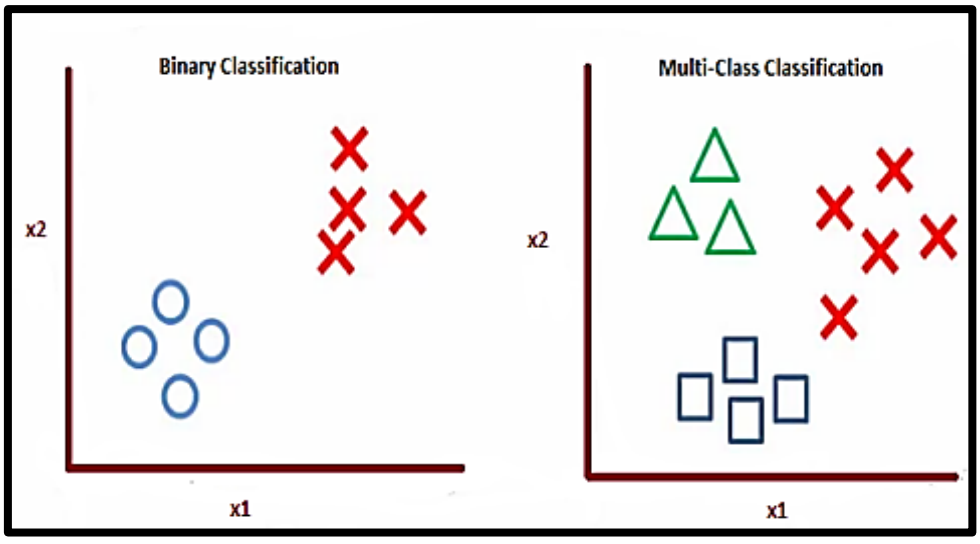


Figure 4: Binary classification vs. Multi-Class Classification

5.1.2 Regression techniques

“These techniques predict continuous variables, such as temperature variations or fluctuation in energy demand. "[8] These techniques are used if we have a range of data and if the results of our work are real numbers.

5.2 Unsupervised Learning

This method or technique does not require you to oversee the model or share labeled data with the model. Instead, the model's algorithm automatically understands the data and begins to learn from it without guidance. The model uses the untagged data to identify new patterns and information based on the design of their algorithm. With this method we can find new and previously unidentified information.

This type of learning behavior is similar to that of humans”. Imagine how we analyze and observe the environment to collect the data and understand and see things. Similarly, machines with unsupervised learning algorithms reveal patterns to find useful results. For example, the system can tell the difference between cats and dogs by understanding the characteristics and characteristics of both animals.

Unsupervised learning models are used for three main tasks: clustering, association and dimensionality reduction:

5.2.1 Clustering

"Clustering is the most popular unsupervised learning technique. Clustering is used to perform exploratory analysis of data to find hidden patterns or clusters in the data." [14] Clustering consists of dividing the population into groups, the data points of the same group are the most similar compared to the other groups. For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the

grouping and granularity. This technique is helpful for market segmentation, image compression, etc.

5.2.2 Association

Exploring association rules finds interesting associations and relationships between large sets of data elements. This rule indicates how often an item set occurs in a transaction. These methods are frequently used for market basket analysis and recommendation engines, along the lines of "Customers Who Bought This Item Also Bought" recommendations.

5.2.3 Dimensionality reduction

is a learning technique used when the number of features (or dimensions) in a given dataset is too high, these dimensions are represented as columns, and the goal is to reduce the number of them.
There are two main categories of dimensionality reduction:

- Feature Selection → we select a subset of features of the original dataset.

- Feature Extraction → we obtain information from the original set to build a new feature subspace.

5.3 reinforcement learning

Reinforcement learning is defined as a type of learner in which new behavior and new experiences are shaped by the positive and negative consequences of our behavior. Each individual pursues the goal of maximizing the positive consequences of his behavior and minimizing the negative ones by choosing the behavior that is most likely to have a positive consequence, depending on the specific environmental situation. Reinforcement learning also

refers to the process of optimizing the selection of discrete behaviors in each specific behavioral situation.

6 Key differences between supervised ML, unsupervised ML and reinforcement ML

-The following table presents the different characteristics of the supervised and unsupervised and reinforcement approaches.

Table 1: Comparison Table

Criteria	Supervised ML	Unsupervised ML	Reinforcement ML
Definition	Learns by using labelled data	Trained using unlabelled data without any guidance.	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No – predefined data
Type of problems	Regression and classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra supervision	No supervision	No supervision
Algorithms	Linear Regression, Logistic Regression, SVM, KNN etc.	K – Means, C – Means, Apriori	Q – Learning, SARSA
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of action
Application	Risk Evaluation, Forecast Sales	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

-The following table presents the advantages and disadvantages of the supervised and unsupervised approach:

Table 2: Advantages and disadvantages of the supervised and unsupervised and Reinforcement approaches

Types of machine learning	Advantages	Disadvantages
Supervised ML	-allows you to be more precise on the definition of labels. -Allows you to determine the number of class we want to have. -The input data is very known and narrow. -The results obtained by the supervised method are more precise and reliable. -The answers in the analysis and the output of your algorithm are likely to be known because all the classes used are known.	-Supervised learning can be a complex method compared to the unsupervised method. -This does not take place in real time unlike unsupervised learning. -If the data is dynamic, voluminous and growing, you will not be sure of the labels used to predefine the rules.

Unsupervised ML	- Less complexity. - Takes place in real time. - It is often easier to obtain data without a label.	- Less precision. - The results of the analysis cannot be determined.
Reinforcement ML	-Reinforcement Learning is used to solve complex problems that cannot be solved by conventional techniques. -This technique is preferred to achieve long-term results which are very difficult to achieve. -This learning model is very similar to the learning of human beings. Hence, it is close to achieving perfection.	-Too much reinforcement learning can lead to an overload of states which can diminish the results. -This algorithm is not preferable for solving simple problems. -This algorithm needs a lot of data and a lot of computation. -The curse of dimensionality limits reinforcement learning for real physical systems.

7 K-fold cross validation

Cross-validation helps evaluate **machine learning** models. This statistical method helps to compare and choose the model in applied **machine learning**. Understanding and implementing this predictive modeling problem is easy and straight forward. This technique has less bias in estimating the model's capabilities. Now let’s understand the concept of k-times cross-validation and how to evaluate a **machine learning** model using this technique. The k-fold cross-validation means that the data set is divided into a number of K. It divides the data set at the point that the test set uses each fold. Let's understand the concept using 5x cross-validation, or K + 5. In this scenario, the method divides the data set into five convolutions. The model uses the first fold in the first iteration to test the model. It uses the remaining data sets to train the model. The second fold helps in testing the data set and the others aids in the training process. The same process repeats until the test record uses each of the five folds.

In addition to the numerous advantages of **machine learning** algorithms, the model follows the same model to predict and generate the data with discrete or continuous values. It is important to ensure that the model's data is accurate and not under- or over-fitted. Underfitting and overfitting are two important terms in **machine learning**. These terms define how well trained a model is to predict data. To check the performance and behavior of the algorithm, the overfitting includes a hyperparameter value.

7.1 Underfitting in machine learning

The model can generate accurate predictions with new data if the model fits the dataset perfectly. A suitable algorithm for the trained data set can help in training the new data set. However, if the **machine learning** model relies on an improper training process, it will not generate accurate data or adequate predictions. As a result, the model will not be able to process important patterns from data sets.

If the model stops during the training process, it will lead to underfitting. This indicates that the data is taking more time to be fully processed. This affects the model's performance for new data. The model will not give accurate results and is unusable.

7.2 Overfitting in machine learning

Overfitting is just the opposite of underfitting. This means that the model not only learns the data and extracts the pattern, but also learns more than it can. This condition suggests that the data is picking up noise, causing the model to generalize to new data. The noise is the irrelevant data that affects the output of the prediction as it encounters new data.

8 Brief description of popular algorithms

8.1 Random forest

Random Forest is a supervised learning algorithm. As you can already see from its name, it creates a forest and makes it random. The "forest" he builds is a set of decision trees, most of the time formed with the "bagging" method. The general idea of the bagging method is just a combination of learning models improving the overall result.[31]

To put it in layman's terms: Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. A big advantage of Random Forest is that it can be used for classification and regression problems, which make up the majority of **machine learning** systems today. Below you can see what a random forest with two trees looks like:

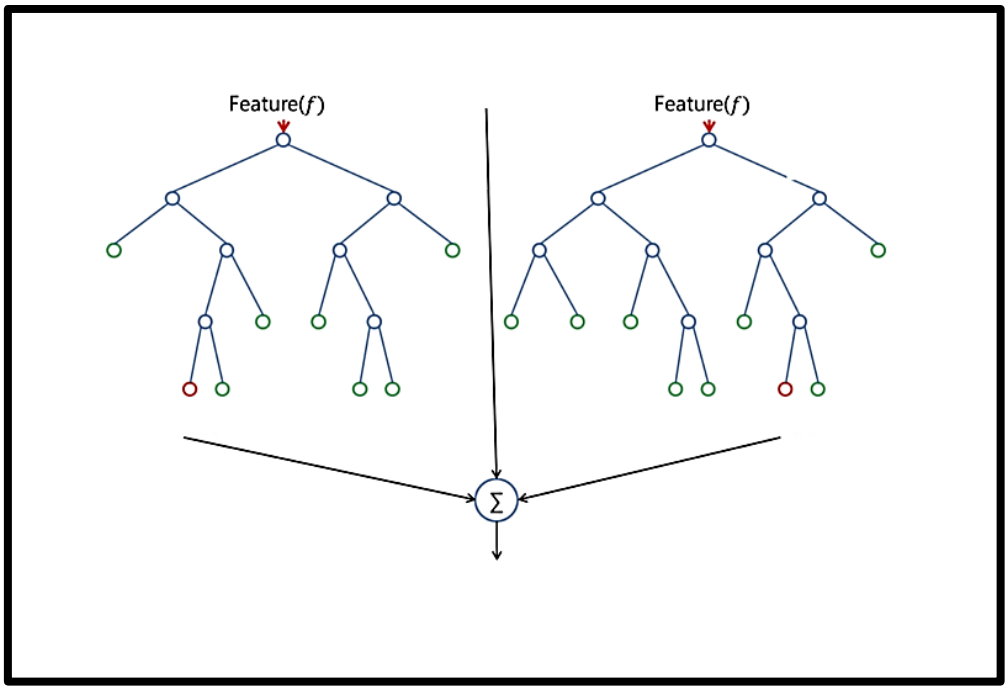


Figure 5: Random forest

8.2 SVM

Support Vector Machine: SVMs are a family of **machine learning** algorithms that solve both classification, regression and anomaly detection problems. They are known for their strong theoretical guarantees, their great flexibility and their ease of use even without great knowledge of data mining. SVMs were developed in the 1990s. As shown in the figure below, their principle is simple: they aim to separate data into classes using a border that is as "simple" as possible, such as: so that the distance between the different groups of data and the border which separates them is maximum. This distance is also called “margin” and SVMs are thus called “wide margin separators”, the “support vectors” being the data closest to the border.

This notion of border assumes that the data are linearly separable, which is rarely the case. To overcome this, SVMs often rely on the use of "kernels".

the kernel functions map the nonlinear separable input space into a larger linear separable feature space and in this new higher dimensional separable linear space, the support vector machines can operate normally. The kernel method then returns the solutions, so that in the nonlinear separable input space you have a nonlinear solution.

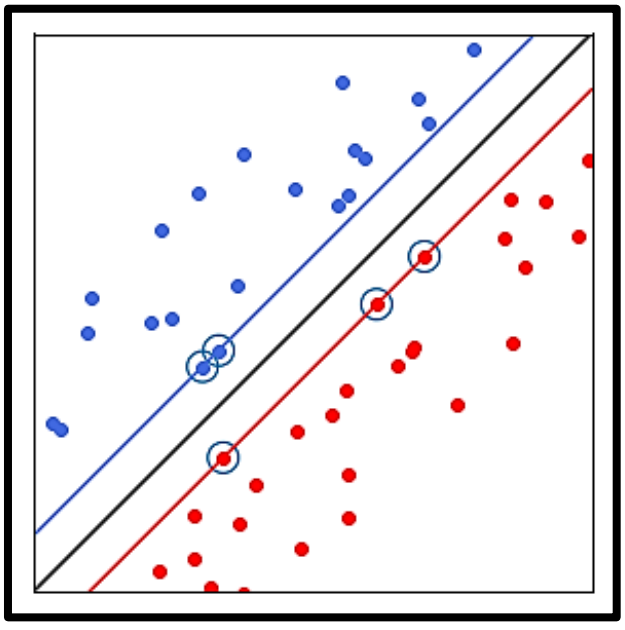


Figure 6: Support vector machine

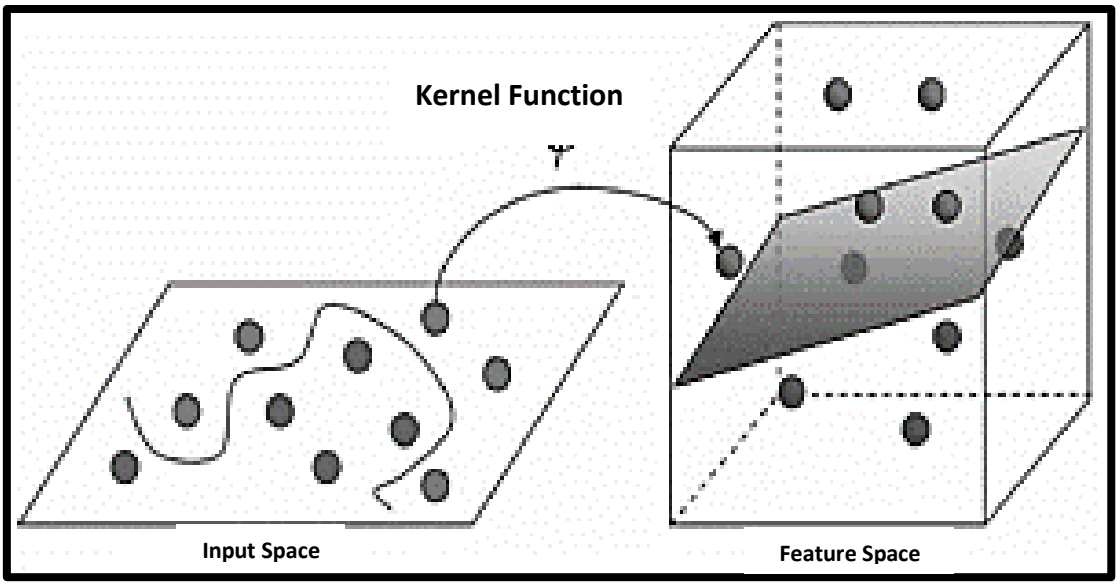


Figure 7: kernel function

In the example above, we have a nonlinear two-dimensional feature space. With the kernel function, we can map the input space into a three-dimensional space. In this feature space, we can then separate the linear learning set. When we restore the solution to the input space, we get a nonlinear solution.[32]

8.3 Neural Networks

Neural networks are another **machine learning** algorithm and they have seen times of great popularity and times when they were rarely used. The first example of a neural network was called the perceptron, which was invented by Frank Rosenblatt in 1957. The perceptron is a network made up of only an input layer and an output layer. In the case of binary classifications, the output layer has only one neuron or unit. The perceptron looked very promising early on, although it quickly realized that it could only learn linearly separable patterns. For example, Marvin Minsky and Seymour Papert have shown that they cannot learn the XOR logic function.

In its most basic representations, perceptrons are just simple representations of a neuron and its input (input which can be made up of several neurons). Given the different inputs in a neuron, we define an activation value by the formula $(a(x)=\sum_i wi\ xi)$ where xi is the value for the input neuron, while wi is the value of the connection between neuron i and the output. If the activation value, which should be considered as the internal state of the neuron, is

greater than a fixed threshold b, then the neuron will activate, that means it will be triggered otherwise it will not be the case. Perceptrons share many similarities with logistic regression algorithms and are also constrained by linear classifiers.

Advantages and disadvantages of neural networks

Table 3: Advantages and disadvantages of neural networks

Advantages	Disadvantages
(a) Use of qualitative or quantitative variables, but better result for the latter (b) Use for classification and regression; (c) Treatment of unstructured problems, without any prior information. (d) No need for independence between variables. (e) No problem with incomplete or noisy data.	(a) The order of presentation of incoming data is important; (b) Need for categorical data encoding; (c) Sensitivity to local minima (d) Ambiguity of their operation, which prevents choosing the best structure adapted to a given problem.

9 One-hot encoding representation

In the data processing step of **machine learning**, we often need to prepare our data in certain ways before we feed it into a **machine learning** model. One of the examples is performing one-hot coding on categorical data. Hot coding is a process in computing that is applied to categorical data in order to convert it to a binary vector representation for use in **machine**

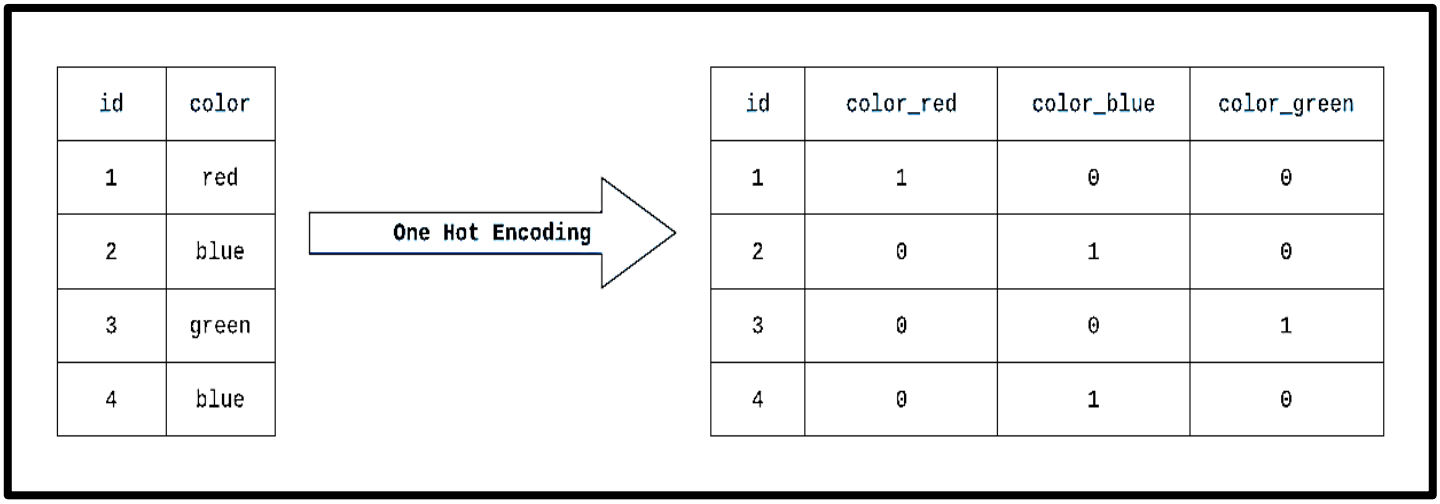


Figure 8: Building a one hot encoding layer

learning algorithms. In one-hot coding, the representation of binary vector arrays enables a

machine learning algorithm to take advantage of the information contained in a category value without the confusion caused by ordinality.

10 Mapping the human brain

Neural networks are processing systems, of which the design is inspired by the structure and functioning of the human brain. Basically, we are trying to replicate the human brain by creating an artificial human brain.

The human brain is principally composed of about 10 billion neurons or what we call nerve cells, these neurons are transmitting and processing the information received as electrical signals from our senses. Each is connected to about 10,000 other neurons, clearly it comprises of a large number of neurons approximately 10^{11} neurons interconnected with each other.

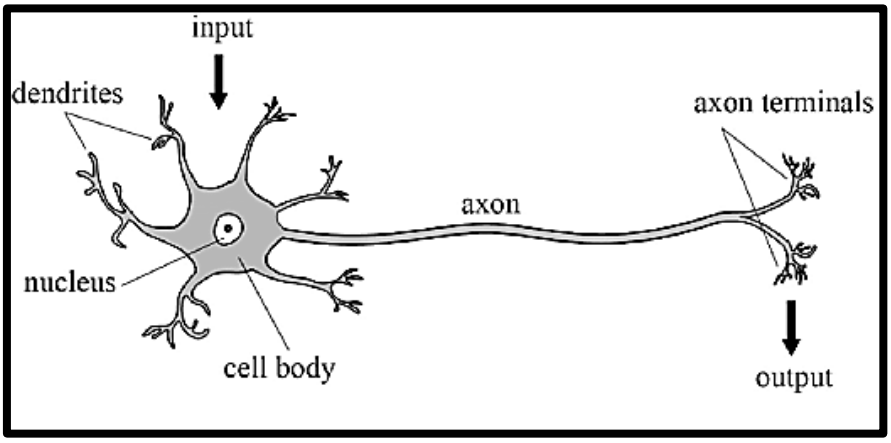


Figure 9: the biological neuron cell

- Dendrites: **they are tree like structural networks made up of nerve fibers. They are connected to cell body.**
- Axon: **it is a single, long connection extending from the cell body. It carries signals from the neuron.**
- Cell body: **it is the main structural part of the neuron which carries the nucleus.**

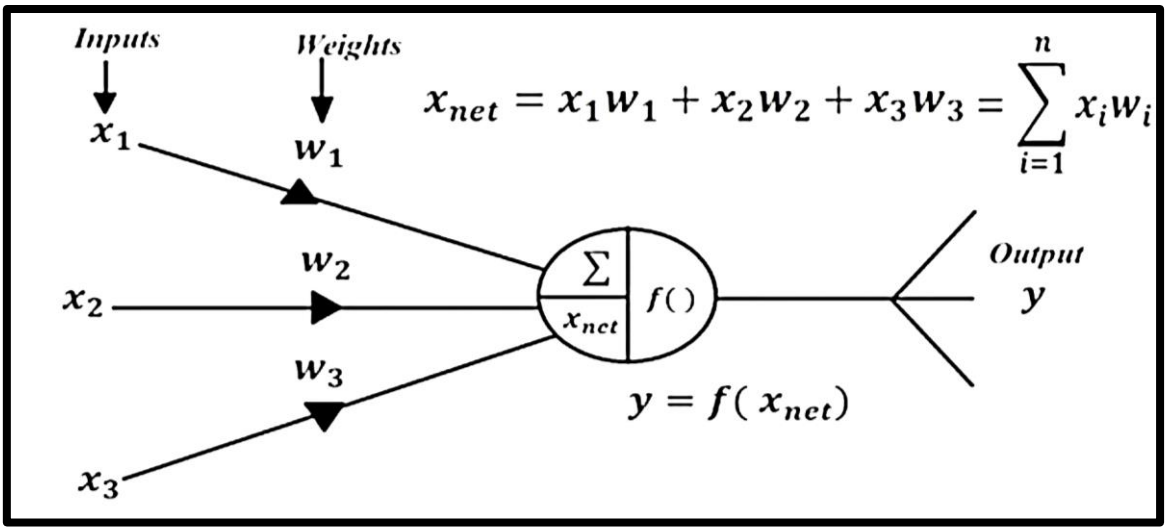


Figure 10: artificial neural network

An artificial neural network works in a similar way: Here a neuron is a mathematical formula that processes an input and generates an output from it. The values of the formula are defined by the output data. Many artificial neurons work together to form an artificial neural network (ANN).

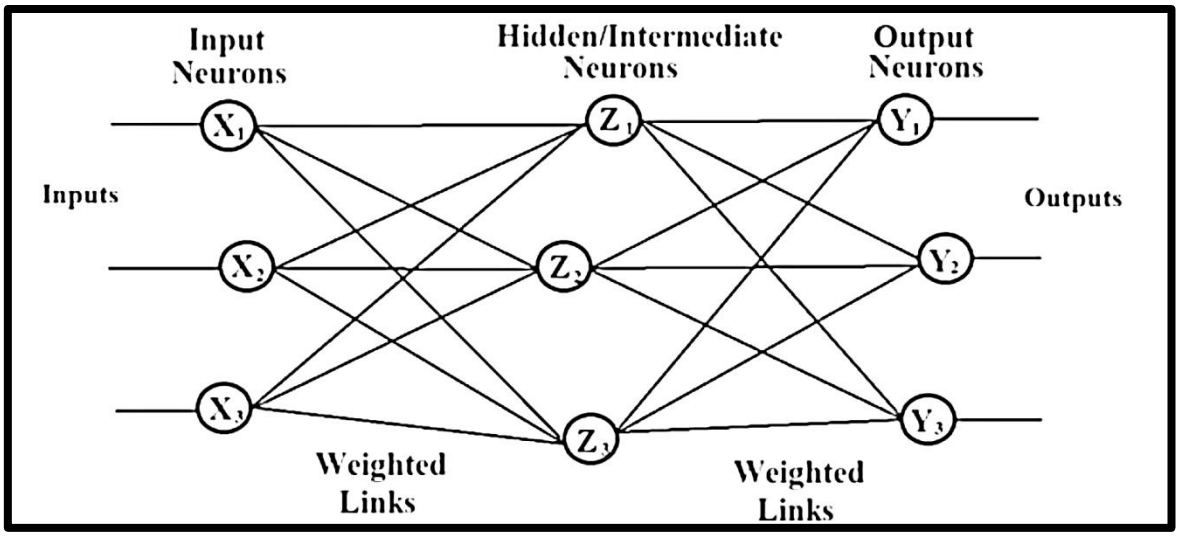


Figure 11: a simple neural network structure

11 Models used in my project

11.1 the Generalized Naive Bayes model

naive Bayesian classifiers assume that the individual features are independent of one another concerning the class variable.

The Bayesian network seeks to determine, P (H / X), the probability of verification of H after observing X. Using Bayes' theorem.

$$P(H|X) = [P(X|H).\ P(H)]\ |P(X)$$

such as:

- X: unlabeled data.

- H: the hypothesis; X belongs to class C.

- P (H / X): probability of X belonging to class C.

- P (H): the probability of occurrence of class C in the population.

- P (X): the probability of occurrence of attribute X in the population.

- P (X / H): probability of appearance of each value of the attributes of X in the attributes of samples belonging to class C:

$$P(X|H) = \Pi P\left(\overline{A}_i = \overline{V}_i|H\right)$$

$\overline{A_i}$attribute of X and $\overline{v_i}$ its value

Despite their simplicity they have strengths:

- **They need a small amount of training data.**
- **They are very fast compared to other classifiers.**
- **They work well for spam filtering and document classification. [41]**

11.2 the K-Nearest Neighbors

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. Some advantages:

- **The algorithm is simple and easy to implement.**

- **There's no need to build a model, tune several parameters, or make additional assumptions.**
- **The algorithm is versatile. It can be used for classification, regression, and search. [36][34]**

11.3 XGBoost Model

XGBoost is a decision-tree-based ensemble **Machine Learning** algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

There are many advantages of XGBoost, some of them are mentioned below:

- It is Highly Flexible
- It uses the power of parallel processing
- It is faster than Gradient Boosting
- It supports regularization
- It is designed to handle missing data with its in-build features.
- The user can run a cross-validation after each iteration.
- It Works well in small to medium dataset.[40]

11.4 Decision Tree Model

Decision Tree models are created using 2 steps: Induction and Pruning. Induction is where we actually build the tree i.e set all of the hierarchical decision boundaries based on our data. Because of the nature of training decision trees, they can be prone to major overfitting. Pruning is the process of removing the unnecessary structure from a decision tree, effectively reducing the complexity to combat overfitting with the added bonus of making it even easier to interpret. [37]

11.5 Random Forest Model

Random Forest is a supervised **machine learning** algorithm that can be used to perform both regression and classification tasks in data mining. It is a set-based technique that can be used

to perform classification. It uses a number of classification trees (like decision trees) and then gives the final result. [42]

11.6 Logistic Regression Model

Simple linear regression is used to find a relationship of one (continuous) output variable to another. his strengths:

- **Easy to understand and explain.**
- **Useful for data analysis.[43]**

11.7 Convolutional neural network

belongs to the family of Artificial Neural Networks which are computational models inspired by the characteristics of biological neural networks

One of the most popular neural networks is the Convolutional Neural Network also known as CNN or Convnet. It takes this name from mathematical linear operation between matrixes called convolution. CNN have multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully connected layer. The convolutional and fully- connected layers have parameters but pooling and non-linearity layers don't have parameters. The **CNN** has an

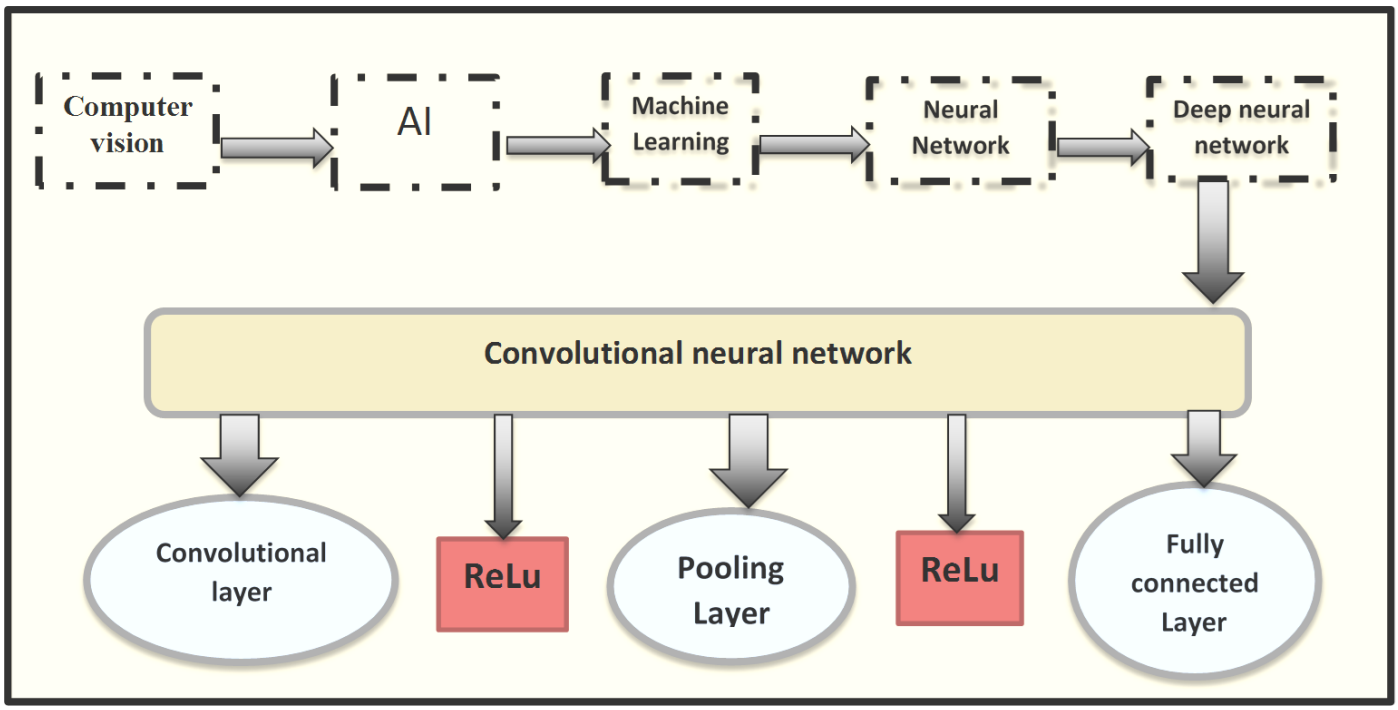


Figure 12: Convolutional Neural Networks

excellent performance in **machine learning** problems. Specially the applications that deal with image data, such as largest image classification.

The most beneficial aspect of CNNs is reducing the number of parameters in ANN. This achievement has prompted both researchers and developers to approach larger models in order to solve complex tasks, which was not possible with classic ANNs. The most important assumption about problems that are solved by **CNN** should not have features which are spatially dependent. In other words, for example, in a face detection application, we do not need to pay attention to where the faces are located in the images. The only concern is to detect them regardless of their position in the given images. Another important aspect of **CNN**, is to obtain abstract features when input propagates toward the deeper layers. For example, in image classification, the edge might be detected in the first layers, and then the simpler shapes in the second layers, and then the higher-level features.

Convolutional neural networks differ from other forms of artificial neural networks in this sense. Instead of focusing on the entire problem area, knowledge about the specific type of entry is tapped; this, in turn, allows for much more efficient network architecture.

Now, let's understand dimensions in **CNN**s.

As we already seen, a convolution requires a kernel, which is a matrix that moves over the input data and performs the dot product with the overlapping input region, obtaining an activation value for every region.[56]

A kernel represents a pattern, and the activation represents how well the overlapping region matches that pattern

The objects that get affected when the **CNN**'s dimension differs are the input and output layers, the kernel and convolution also (in what dimensions the kernel can move).

In Conv1D the input and output data are 2 dimensional. The kernel moves in one direction. Mainly used for time series.

In the Conv2D, the input and output data are 3 dimensional, and the kernel moves in 2 directions. And it is used for image data.

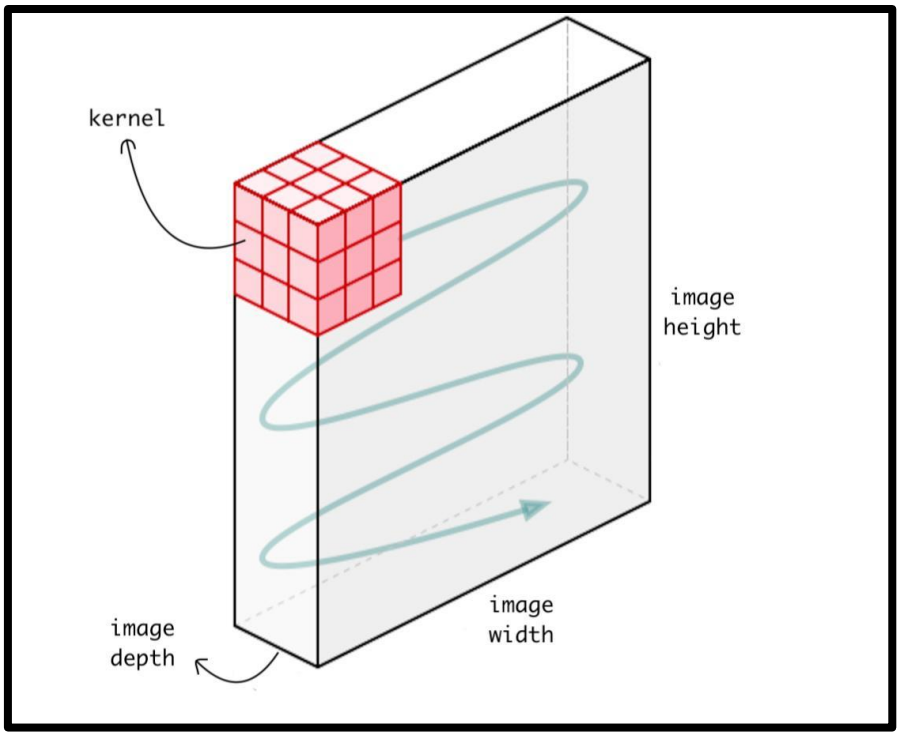


Figure 13: Two-dimensional **CNN** | Conv2D

In Conv3D, the input and output data of 3D **CNN** is 4-dimensional. Usually used for 3D image data (MRI, CT scan).

12 Conclusion

I hereby devoted this chapter to the presentation of basic concepts such as **machine learning** and its types, and I can tell now that my classification is binary.

in conclude with the most popular and most used algorithms and by specifying **CNN** algorithm.

Chapter 3

Related Work

1 Introduction

As we seen in previous Chapters, phishing attacks have gained a lot of energy all around the world lately and the number of phishing detections have increased. For that it became a good reason for researchers to announce the war against phishing so that fighting phishing becomes a success story for real-world **machine learning**.
In this chapter, I present the most relevant approaches, with some examples of related work to each approach. The techniques known in the classification method are:

- **Blacklist and whitelist.**
- **Heuristics.**
- **Visual similarity.**
- **Machine learning.**

2 Various anti-phishing approaches:

In order to prevent phishing attacks scientists have shown so much efforts so that the latter stop increasing and affecting more organizations and individuals by developing different methods, these methods have become an obligation and have been proposed to solve the problem of phishing attacks.

2.1."Blacklist / whitelist" based solutions

A blacklist contains the URL addresses of known phishing sites. The blacklist solution is generally deployed as a toolbar or an extension of web browsers. As examples of tools implementing this type of solution we find: Mozilla Firefox, googlesafebrowsing and phishtank [23]. In a whitelist approach, all legitimate websites are listed; any website that is not in the list is recognized as invalid and causes a warning to the user. It is impractical to create a global whitelist of the entire World Wide Web due to the large scale and rapid increase in volume. But an individual whitelist is possible. A common user can only connect to a limited number of websites, AIWL (Automated Individual White-List) uses this feature to

create an individual whitelist that registers all familiar interfaces of the user's websites to effectively defend against phishing attacks. [49]

2.2. Heuristics-based solutions

Heuristic development is a method of solving problems that does not rely on the detailed examination of the said problem. This consists of working by successive approaches. For example, by highlighting similarities with difficulties already encountered, in order to gradually eliminate the alternatives and keep only a sample of solutions. The goal will be to keep the one that is optimal in relation to the problem encountered. In "the modeling of complex systems" (1991), Jean-Louis Le Moigne (famous French specialist in epistemology and systems) gives this explanation: "A heuristic is a formalized reasoning of problem solving (representable by a computation known) which is held to be plausible but not certain that it will lead to the determination of a satisfactory solution of the problem ". Although this heuristic method is used to spot new viruses, as well as versions of already known viruses, the efficiency remains relatively low, considering the number of false positives. Like biological viruses, computer viruses are constantly changing and evolving. [50]

2.3. Method based on visual similarity

The main purpose of phishing is to deceive the user by crafting an accurate image of the legitimate site so that the user has no suspicion of the phishing site. Therefore, the anti-phishing techniques compare the image of a suspicious website to a legitimate image database to get the similarity ratio, which is used for the classification of suspicious websites. The website is classified as phishing when the similarity score is above a certain threshold, otherwise it is considered legitimate. [51]

2.4. Machine learning

These approaches attempt to analyze information from a URL and the corresponding website or web page, extracting good representation from URLs, and training a prediction model on the learning data of malicious and non-malicious URLs. [51]

3 Related work based on machine learning

Several approaches have been proposed in the field of malicious URL detection. Most of this research has led to improvements in this area by using the different classification algorithms such as: SVM, KNN, Decision tree, Naïve Bayes, etc.…

I mention below some works in this area:

Mouhuidin Ahmed, Abdunaser Mahmoud carried out an approach which analyzes abnormal behavior in a computer to designate a suspicious attack on a system, the approach then uses classification techniques (SVM, KNN, neural networks, etc.) for the detection of 'anomaly. [52]

S. Jagdessam and aI [53]. this approach only uses URL information to determine whether a website is legitimate or not. The user does not have to visit the website to determine the ranking of the site as it has been done before through the use of the Random Forest algorithm which provides this advantage.

In Dessa and aI [54] created a chrome extension as a middleware between user and website, the downside of this work is that harmful content cannot be collected in mass exhaustively and therefore this approach is limited, it does not cannot detect phishing off url.

S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang [23] studied the effectiveness of phishing blacklists. This detection method extracts the sender's address and link address in the message and checks whether it is in the blacklist to distinguish whether the email is a phishing email. The update of a blacklist is usually reported by users, and whether it is a phishing website or not is manually identified. the perfection of the blacklist determines the effectiveness of this method based on the blacklist mechanism for phishing email detection.

Ankut Kumar and Gupta [55] this approach uses the logistic regression algorithm which extracts functionality only from the client side which is based on HTML.

4 conclusion

In this chapter we have presented the different methods of detecting malicious websites then we have presented an overview of the work that has been done to classify web pages. This study allowed us to know a panoply of techniques, we chose that of **machine learning** based on the combined characteristics of the URL and the HTML file, this technique will be the subject of our study in the next chapter.

Chapter 4

Proposed approach

1 Introduction:

In the light of the previous studies that we did about various anti-phishing approaches that several researchers worldwide propose, I've focused in the last chapters on a specific category that has a potent technique that detects phishing with high accuracy and precision. I am talking about **machine learning**-based methods. Those methods that use different classifiers achieved the best results and proved that it is a powerful tool to break down phishing attacks.

Phishers are constantly creating new attacks and fraudulent techniques to hack anti-phishing tools to get what they want, and as we mentioned earlier, they use systematic methods to achieve that. So, it becomes a necessity to catch up with phishing trends and reports and update our knowledge in order to adjust our anti-phishing techniques and tools.

Following that, we choose to concentrate on the techniques that used **CNN** classifiers to detect phishing, in intent to adjust this kind of technique and to propose a robust URL-based approach that gives accurate results that decide whether a URL is phishing or not.

For the purpose to get a robust URL based approach that gives the best results and precision

2 Proposed approach:

As a first step, an analysis of previous work in the field of phishing detection allowed us to examine the different types of attacks and the approach implemented for the detection in my approach I have built a vector of attributes to describe the page visited. This vector is subsequently used by a **CNN** classifier to decide whether the page is legitimate or not. In my approach, I use 30 attributes. This project is mainly to implement the mentioned model in python. The **CNN** Classifier should be trained on the phishing website dataset using python scikit-learn, then the learned model should be saved by using pickle and then deployed to the website I create by using HTML and CSS and our saved learned model.pkl by using the instruction pickle.dump. to deploy the model I had to use flask micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[3] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

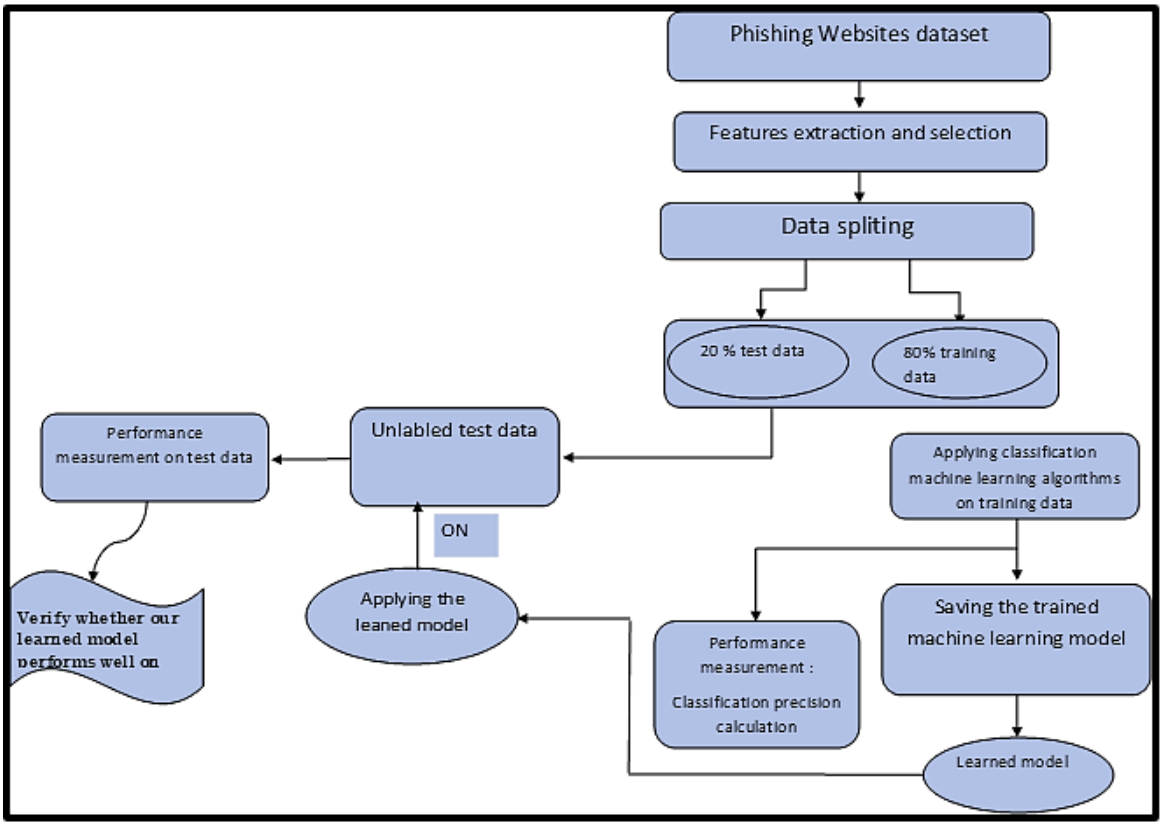


Figure 14: Approaches' Architecture

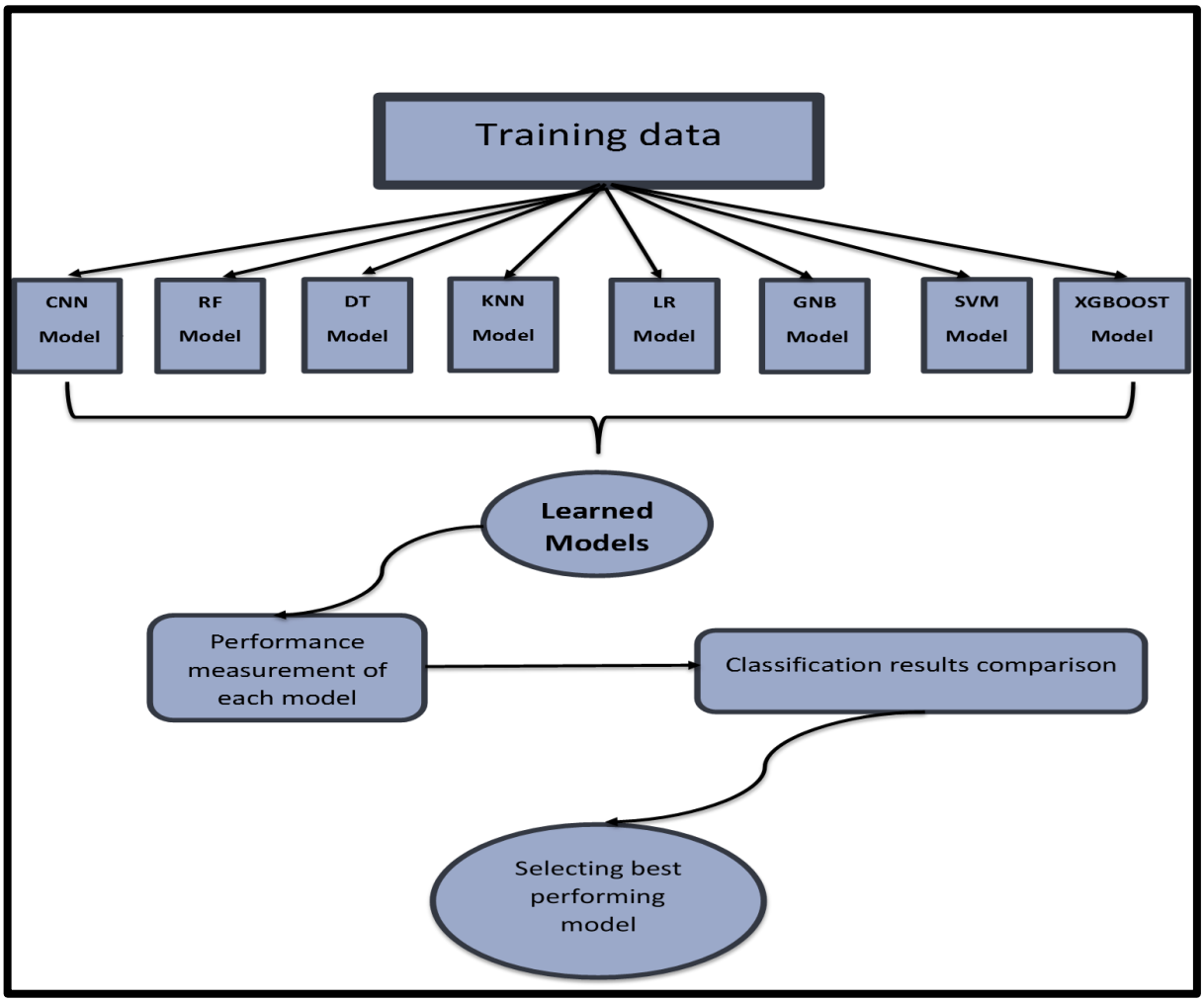


Figure 15: Selection of the best performing model

3 Dataset features

Table 4: View of the dataset

	having_IP_Address	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_Sta
0	2	1	1	1	2	2	2	2
1	1	1	1	1	1	2	0	1
2	1	0	1	1	1	2	2	2
3	1	0	1	1	1	2	2	2
4	1	0	2	1	1	2	1	1
...	...	...	...	...	...	...	...	...
11050	1	2	1	2	1	1	1	1
11051	2	1	1	2	2	2	1	2
11052	1	2	1	1	1	2	1	2
11053	2	2	1	1	1	2	2	2
11054	2	2	1	1	1	2	2	2

a. URL and derived features

These features are included in the address of the website. Phishers generally adopt following methods to attack:

1.Long URL: long URLs are commonly used by phishers to hide the part that is hesitant in the address bar. For that we check the URL length if it is greater than or equal to 54 characters then It's classified as a phishing URL else it's legitimate.

2. Providing IP instead of URL: using IP address instead of domain name in the URL is a proof that someone is trying to steal your personal information by redirecting victims to phishers machine so the rule is if the domain part has an IP address than it's phishing otherwise it's legitimate.

3. Using shortened URLs: URL shorteners grew up in popularity ad that's what made phishers begin to exploit URL shorteners with phishing scams, so we are going to consider any TinyURL as phishing otherwise it's legitimate.

4. "@" symbol in URL: the symbol "@"ignores anything preceding it so URLs with "@" are taken as phishing ones.

5. URLs with "//": "//" equals to redirecting the user to a different website. So we should calculate which position the "//" is placed by putting in consideration ("http:" or "https:") if it's position is greater than 7 then it's phishing else it's legitimate.

6. URL with "-": Legitimate websites rarely use "-". However, phishing websites use "-" in URLs to mimic the names of legitimate websites.

7. Number of subdomain: Legitimate websites generally use no or only one sub-domain, however, phishers generally redirect via multiple sub-domains.

8. Use of HTTPs security: HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses. As a result, HTTPS is far more secure than HTTP, but it was never enough to fight phishing , researchers have suggested to check the age of the certificate assigned with https including the trust issuer so if a URL is using HTTPS and issuer trusted and age of certificated is greater than or equals to 1 it's legitimate if issuer not trusted suspicious else phishing.

9. Period for which Domain has been registered: Legitimate websites usually operate over several years. Most phishing websites operate for a short period of time and do not have domain registered for more than one year.

10. Favicon: If the favicon is loaded from a domain other than that shown in the address bar, then the webpage is likely to be considered a Phishing attempt.

11. Ports: All websites running over HTTP use port 80 and running over HTTPS use port 443. The other ports should remain closed for security reasons. However, when we inspected some popular websites like www.google.com, www.linkedin.com, www.yahoo.com, etc. we found most of them have their FTP (21), SSH (22) and other non-standard ports open. Hence, we decided to drop this feature.

12. Use of "https" in domain part: Phishers Use HTTP Token in domain part of the URL to trick users into believing that the URL passes through secure "HTTPS" protocol. And this is taken as phishing otherwise legitimate

b. Page's source code-based features

A common trick employed by phishers is to make the interface textually and graphically very similar to a legitimate webpage. However, these features are difficult to tap especially when the content is loaded dynamically. However, there are a number of distinguishing features between legitimate and phishing websites based on code structure of the webpage: Based on URLs embedded in webpage The URLs being accessed/accessible by the webpage generally carry a good amount of information about their nature. If the links belong to the website itself,

it increases the credibility of the website. Few features identified on the basis of the embedded URLs are:

1. Embedded objects' URLs: Legitimate pages share their domains with the objects embedded in them. However, phishing websites usually load embedded objects from external resources to resemble them.

2. URL of Anchor tag: Anchor tag in HTML is used for hyperlinking. if percentage of URL of anchor s less than 30 percent than legitimate,

If it's between 31 and 67 it's suspicious else it's phishing.

3. Tags: Legitimate pages have the domain name for the page and domain name of URLs in it's <Meta>, <Script> and <Link> tags as same. However, they usually differ for suspicious websites.

% of Links in "<Meta>","<Script>" and "<"Link>\""<17% → Legitimate

% of Links in <Meta>","<Script>" and "<"Link>\"" ≥17% And≤81% → Suspicious

Otherwise→ Phishing

4. Server Form Handler (SFH): Legitimate websites always take action on the content submitted via a form.

SFH is ""about: blank\"" Or Is Empty → Phishing

SFH "Refers To " A Different Domain→ Suspicious

Otherwise → Legitimate

5. Submitting information to a mail: Legitimate websites generally process the information submitted either on frontend itself or on some backend. However, a phisher might redirect the information to his personal mail.

6. Abnormal URL: The host name is usually included in the URL of all the objects on webpage. However, since this feature was already well explored by other factors above, we decided to drop it.

c. HTML and Javascript based features

HTML and Javascript can be used to hide malicious code in a seemingly well-behaved website. Some of the features recognised are:

1. Number of times website redirects: see how many times a website has been redirected

Of Redirect Page ≤1 → Legitimate

of Redirect Page ≥2 And <4→ Suspicious

Otherwise → Phishing

2. Status bar customization: Phishers generally use Javascript to modify the URL of the webpage as seen in the address bar and it is mostly different from the actual URL.

onMouseOver Changes Status Bar→ Phishing

It doesn't Change Status Bar → Legitimate

3. Right-click disabled: Right click can be used to view the source code of a webpage. However, to eliminate this risk of being caught, phishers generally disable right click.

4. Pop-Up windows: If legitimate websites, use pop-up window, it is mostly for an alert purpose. However, phishing websites usually collect information through pop-up windows.

However, it seemed infeasible to detect pop-up windows without actually visiting the webpage. Hence, we decided to drop this feature.

5. IFrame redirection: Phishers can overlap a webpage

with an invisible frame and redirect to a different website/server using it.

Using iframe→ Phishing

Otherwise → Legitimate

d. Domain based features

The domains of legitimate websites are generally established for long duration and have good statistical properties.

However, phishing websites usually live for shorter time periods and do not earn good indicators.

1. Age of the domain: Legitimate domains have a minimum age of 6 months. However, the phishing websites are shot-lived.

2. DNS: Legitimate sites are mostly recognized by publicly available WHOIS database, and have a non-empty DNS record. Phishing websites are mostly not recognized by WHOIS database.

no DNS Record for The Domain → Phishing

Otherwise→ Legitimate

3. Website Traffic: Legitimate domains have a lot of visitors, hence are ranked among the top 100,000 in Alexa database. However, if a website is not even recognized by Alexa, then it is deemed to be phishing.

4. Page rank: Legitimate domains generally have a page rank between 0.2 and 1. Higher page rank implies that it is an important domain.

5. Google Index: Legitimate websites are usually indexed by Google. Phishing websites generally, being short-lived do not make it to the index.

However, we did not find any publicly available API to use for determining if a website has Google index and thus we dropped this feature.

6. Numbers of links pointing to a page: Legitimate websites generally have a lot of external links pointing to them. But, no reliable source was found from which this information can be extracted. So, we dropped this feature.

7. Statistical Report based: A few publicly available databases like PhishTank are maintained and periodically updated to identify phishing websites. If a website is found in this database labelled as phishing, the probability of being actually phishing is very high.

Host Belongs to Top Phishing IPs or Top Phishing Domains → Phishing

Otherwise → Legitimate.

4 Conclusion

I talked in this Chapter about the proposed approach and its architecture and added a detailed description of the features of my dataset that I am working with.

Chapter 5

Design and Implementation

1 Used Tools

- **Python:**

It is an interpreted programming language, so it does not need to be compiled in order to function. An "interpreter" program lets you run Python code on any computer. This allows you to quickly see the results of a change in the code.

I am using the latest version 3.9.2

```
PS C:\Users\kelto\OneDrive\Bureau\Keltoum_Thesis_CODE> python -V
Python 3.9.2
PS C:\Users\kelto\OneDrive\Bureau\Keltoum_Thesis_CODE>
```

Figure 16: Check Python's current version

- **Weka**

Waikato Environment for Knowledge Analysis (WEKA) is a **machine learning** workbench and open source software that is used for researches; it provides a whole package of **machine learning** classifiers and algorithms to solve many data mining problems such as: clustering, association rule mining and attribute selection, classification, and regression. Also, it offers a lot of modulation and visualization tools such as trees [48].

- **Visual studio code**

Visual Studio Code (later VSC) is an open-source, free, cross-platform (Windows, Mac, and Linux) code editor developed by Microsoft, not to be confused with Visual Studio, Microsoft's proprietary IDE. VSC is developed with Electron and takes advantage of advanced editing features of the Monaco Editor project. Mainly designed for application development with JavaScript, TypeScript and Node.js, the editor can adapt to other types of languages thanks to a well-supplied extension system.

- **WSL**

The Windows Subsystem for Linux 2 (WSL2) is probably the simplest and most efficient solution for using Linux applications under Windows. WSL2 enables access to Linux tools and applications directly from the familiar Windows environment and is therefore particularly interesting for developers.

- **Html**

HTML5 is a markup language used to write websites. HTML5 defines which elements a website contains and which structure it has. A website consists of many so-called HTML5 elements.

- **CSS**

The abbreviation CSS stands for "Cascading Style Sheets", which translates as "stepped design sheets". In short: CSS is used to design websites.

In order to create the design of a website, for example font size, font color and other characteristics, a uniform programming language is required. HTML and CSS are the most commonly used languages to design websites.

CSS prevailed mainly because of its simplicity. CSS is easy to use, clearer than other standards, comparatively easy to learn and allows websites to load faster.

2 Used libraries

When it comes to programming in Python, libraries are where the real power is. Similar to plugins or extensions, libraries unlock your code and make it more efficient. With a few fundamentals of Python programming, you can use libraries for high-performance data analysis and data visualization.

- **Numpy**

NumPy is a library for the Python programming language. NumPy is very popular because it makes writing programs easy. Python is a high-level language, which means you don't have to allocate memory manually. With low-level languages, you have to define memory allocation and processing, which gives you more control over performance, but it also slows down your programming. NumPy gives you the best of both worlds: processing performance without all the allocation. NumPy extends Python with functions for scientific and numerical calculations. The library enables efficient calculations with matrices, multi-dimensional arrays and vectors. [34]

- **Pandas**

Pandas is a python library that allows you to easily handle data to analyze:

Although NumPy provides fundamental structures and tools that make working with data easier, there are several things that limit its usefulness:

- The lack of support for column names forces us to frame questions as multi-dimensional array operations.
- Support for only one data type per ndarray makes it more difficult to work with data that contains both numeric and string data.
- There are lots of low-level methods, but there are many common analysis patterns that don't have pre-built methods.

The Pandas library provides solutions to all of these pain points and more. Pandas is not so much a replacement for NumPy as an extension of NumPy. The underlying code for pandas uses the NumPy library extensively, which means the concepts you've been learning will come in handy as you begin to learn more about pandas.[36]

The primary data structure in pandas is called a dataframe. Dataframes are the pandas equivalent of a Numpy 2D ndarray, with a few key differences:

- Axis values can have string labels, not just numeric ones.
- Dataframes can contain columns with multiple data types: including integer, float, and string.

And this is how to install libraries:

Pip install "name of the library"

```
PS C:\Users\kelto\OneDrive\Bureau\Keltoum_Thesis_CODE> python -m pip install pandas
Collecting pandas
  Downloading pandas-1.2.4-cp39-cp39-win_amd64.whl (9.3 MB)
     |████████████████████████████████| 9.3 MB 57 kB/s
Collecting pytz>=2017.3
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
     |████████████████████████████████| 510 kB 123 kB/s
Collecting numpy>=1.16.5
  Downloading numpy-1.20.3-cp39-cp39-win_amd64.whl (13.7 MB)
     |████████████████████████████████| 13.7 MB 43 kB/s
Collecting python-dateutil>=2.7.3
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
     |████████████████████████████████| 227 kB 261 kB/s
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, numpy, six, python-dateutil, pandas
Successfully installed numpy-1.20.3 pandas-1.2.4 python-dateutil-2.8.1 pytz-2021.1 six-1.16.0
```

Figure 17: The Command used to install any library

- **Tensorflow**

TF is an open source Python library for fast and large numerical computing developed and released by google primarily. It can be used for tasks related to **machine learning** and

artificial intelligence (AI). The framework offers a wide range of possible uses and enables learning neural networks to be created. It is characterized by its good scalability and can be operated on different systems from smartphones to clusters with many servers.[39]

- **Keras**

Keras enables the rapid implementation of neural networks for deep learning applications. It is an open source library written in Python that can be used in conjunction with frameworks such as TensorFlow or Theano.[38]

- **Scipy**

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely.[34]

- **Seaborn**

Seaborn is a freely available library for the Python programming language. The library can be used to visualize data. The library is based on the Matplotlib library and requires other libraries such as NumPy, SciPy and Pandas. Seaborn can be used to turn data into clear graphs and charts. Different diagram types, maps and plots are supported.[34]

- **Matplotlib**

Matplotlib is a plotting library like GNUplot. The main advantage over GNUplot is the fact that matplotlib is a Python module. Due to the growing interest in the Python programming language, the popularity of matplotlib is also increasing continuously. [34]

- **Sklearn**

The free software library Scikit-learn is platform-independent and designed for **machine learning** with the programming language Python. She works with scientific Python libraries such as SciPy or NumPy.

3 Implementation

3.1 The used dataset

The dataset used in this project is provided by the UCI **Machine Learning** repository. The dataset is the result of the work done by Mohammad et al [56]. The dataset is a two-

dimensional matrix containing 11055 rows and 30 columns. Every row of the dataset is a data point that describes a different URL. Every URL is described by 30 features spanning along the columns of the dataset. The 31st column of every row is the decision result (legitimate or phishing URL). A detailed description of the whole features of this dataset is provided in my work in Chapter 4, see the section "3 Dataset Features". In my work, I have made usage of 25 features instead of the whole amount of features. This means that the number of columns in the dataset I am using is 26 (including the decision label), instead of 31. Feature filtering is done manually by choosing the most valuable and promising attributes that may contribute for a better prediction. The set of selected features is shown in Figure 19, specifically in the variable "reduced_df". This variable is a python DataFrame (a 2-dimensional matrix) that contains all the data I am using for the specific list of selected features.

The dataset contains categorical data, i.e., the values of the dataset are not Integers, however, they are defined values (string, character, or integer) of a specific set of user defined values. In my case, the dataset I am studying contains categorical features with values defined in the set of [-1, 0, 1]. Each feature of the dataset is categorized as being legitimate, suspicious or phishing. At a specific URL row, the value -1 means that the URL at a specific attribute column indicates a phishing website. The value 0 means that the URL is a suspicious website. Finally, the value 1 refers to the fact that the URL at a specific attribute is a legitimate website. An example of the content of the targeted dataset is given in Figure 18. Figure 18 shows the head of the dataset, in other words, the first lines of the dataset. This result displayed in the VS Code terminal is provided by the line of code given in Figure 19. The python method "head()" is used for this purpose.

```
   having_IP_Address  URL_Length  Shortining_Service  ...  Links_pointing_to_page  Statistical_report  Result
0                 -1           1                   1  ...                       1                  -1      -1
1                  1           1                   1  ...                       1                   1      -1
2                  1           0                   1  ...                       0                  -1      -1
3                  1           0                   1  ...                      -1                   1      -1
4                  1           0                  -1  ...                       1                   1       1

[5 rows x 31 columns]
```

Figure 18: the head of the dataset

```
def summary_steps(self):
    complete_training = self.getProcessedDataFrame("Dataset/Training Dataset.arff")

    print(complete_training.head())

    #print(complete_training['Result'].value_counts())

    reduced_df = complete_training[['having_IP_Address', 'URL_Length', 'Shortining_Service',
                'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix',
                'having_Sub_Domain', 'SSLfinal_State', 'Domain_registeration_length',
                'Favicon', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor',
                'Links_in_tags', 'SFH', 'Submitting_to_email', 'Redirect', 'on_mouseover', 'RightClick',
                'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
                'Statistical_report', 'Result']]
```

Figure 19: the line of code to get the head of the dataset and to reduce the features

The database I am using is a labeled dataset. This is typical for the fulfillment of a supervised classification task. Each data point is classified either as legitimate or phishing. Consequently, the **machine learning** problem I am trying to solve in my project is a supervised binary classification task.

From the 11055 URL data points, 6157 data points are legitimate URLs, while 4898 are phishing URLs, see Figure 18. The dataset is accompanied by a set of rules to categorize features for any new URL. These rules are defined in [56].

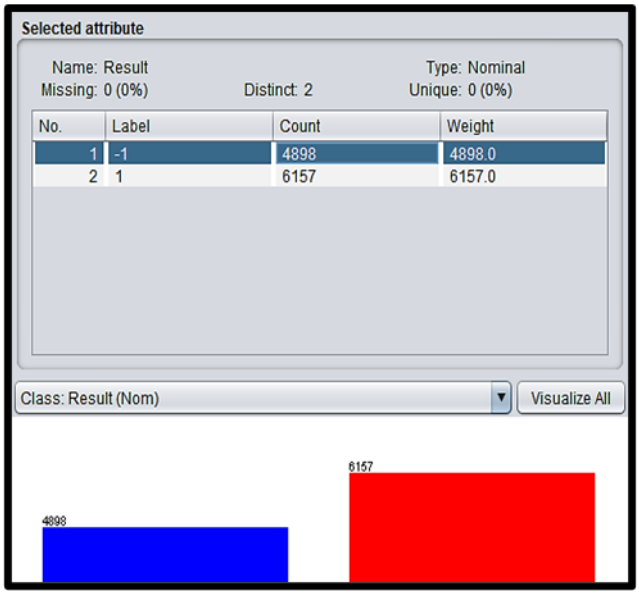


Figure 20: A screenshot from Weka showing the amount of phishing and legitimate websites in the used dataset

3.2 Code structure

Figure 21 shows my VS code workspace. The main folder contains a set of Python classes {**CNN**_Model(), DT_Model(), GNB_Model(), KNN_Model(), LR_Model(), RF_Model(), SVM_Model(), XGBooost_Model()}. Every class contains a set of Python methods required for the generation of a specific ML model. In addition to the **CNN** algorithm, in my work, I have implemented several ML methods to solve the binary classification problem of phishing websites. This is to compare the classification results given by every algorithm and have an idea which algorithm performs better for this type of phishing websites data. A comparative study is conducted to evaluate the performance of every ML approach on the data I am analyzing. In addition to the Python classes, the dataset I am using is saved under the name "Training Dataset.arff" in the folder Dataset of my workspace. The resulting trained models are saved in the folder "Trained_Models". The website is saved in the folder "Extension" of my workspace. The main Python method is in "main_hamaidi.py". In this Python file, all the ML classes are called and the results are summarized. The Python file "Main_Data_Processing.py" contains the set of Python methods related to loading the data, cleaning and filtering the data, data mapping and splitting.

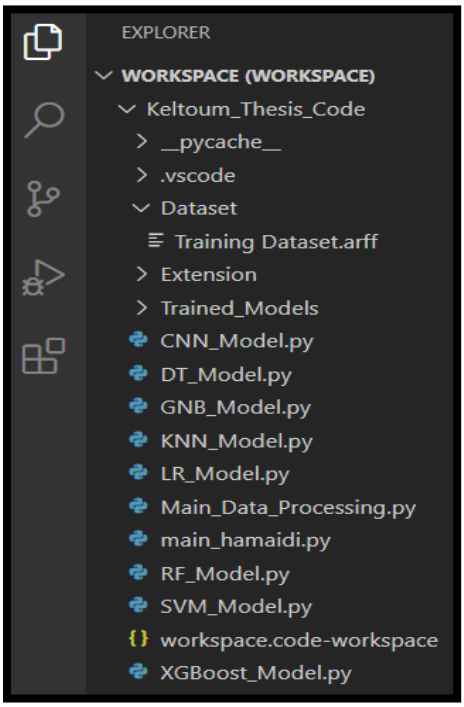


Figure 21: The VS Code workspace and the set of classes

4 Mathematical model for the used performance measures

The performance of the entire system is evaluated using the standard parameters described below.

4.1 Cross validation score

Learning the parameters of a prediction function and testing it on the same data is a methodological error, a model that would only repeat the labels of the samples it just saw would have a perfect score but would not predict anything. this is useful on data that is still invisible. This situation is called overfitting.

To solve this problem, another part of the data set can be presented as a so-called “validation set”: the training takes place on the learning set,

after the evaluation is done on the validation set, and when the experiment seems successful, the final evaluation can be done on the test set. However, by partitioning the available data into three sets, we significantly reduce the number of samples that can be used for training the model, and the results may depend on a particular random choice for the pair of sets (train,validation).

One solution to this problem is a procedure called cross-validation (CV for short). A test set should still be presented for final assessment, but the validation set is no longer required when creating CVs. In the basic approach, called k-fold CV, the training set is divided into k smaller sets (other approaches are described below, but generally follow the same principles). The following procedure is followed for each of the k “folds”:

A model is trained using k folds as training data; the resulting model is validated on the remaining part of the data (i.e., it is used as a test set to calculate a performance measure such as precision).

4.2 confusion matrix

I used in my approach a **CNN** classifier to decide if the URL is legitimate or not, a binary classifier is very suitable for my case since I only have two cases (phishing or legitimate).

The output is either legitimate (+ v) or phishing (-v).

There are only 4 cases where an "X" URL could occur.

• True positive (TP): the prediction is + v and X is legitimate, we want

• True negative (TN): the prediction is – v and X is phishing, we want that too

• False positive (FP): the prediction is + v and X is phishing, false alarm, bad

• False negative (FN): the prediction is – v and X is legitimate, the worst

4.3 Accuracy

It is the ratio of number of correct predictions to the total number of input samples. Since the objective requires most of the URLs to be classified correctly, hence high accuracy is one of the metrics

4.4 False Positive Rate (FPR)

It is the ratio of number of samples incorrectly identified as positive to total number of actually negative samples. The requirement is to minimize the number of phishing websites identified as legitimate as it can lead to heavy losses for the person visiting the website. Thus, low FPR is one of the metrics used.

4.5 Precision

Precision is the ratio of the + v correctly labeled by our program to all the + v labeled.

The precision answers the following question: How many URLs that we have qualified as legitimate are actually legitimate?

Precision = TP / (TP + FP)

4.6 Recall

The recall is the ratio between the + v correctly labeled by our program and all those which are actually legitimate.

Recall answers the following question: Of all the valid URLs, how many are we predicting correctly?

Recall = TP / (TP + FN)

4.7 The F1 score

The F1 score is a measure of the accuracy of a test. It considers both the precision and the recall of the test to calculate the score: the precision is the number of correct positive results divided by the number of all positive results returned by the classifier, and the recall is the number of correct positive results. divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic mean of precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

The F1 score can be interpreted as a weighted average of precision and recall, where an F1 score reaches its best value at 1 and the worst score at 0. The relative contribution of precision and recall to the F1 score is equal. The formula for the F1 score is:

$$F1 = 2 * (precision * recall) / (precision + recall)$$

```
tp, fn, fp, tn = metrics.confusion_matrix(y_train_red[val], clf_xgb.predict(X_train_red_onehot[val])
accuracy = (tn+tp)/(fp+fn+tp+tn)
precision = tp/(tp+fp)
recall = tp/(tp+fn)
fpr = fp/(fp+tn)
f1 = 2*precision*recall/(precision + recall)
accuracy_scores_xgb.append((accuracy, precision, recall, fpr, f1))
```

Figure 22: confusion matrix

5 Implementation and Results

5.1 Support Vector Machine:

```
""" This class is created to implement the SVM Classifier
    Done by: Keltoum Hamaidi """

import numpy as np
from sklearn import metrics
from sklearn.model_selection import GridSearchCV
from sklearn import svm
import pickle
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt

class SVM_Model:

    def plot_cm(self, y_val, y_pred):···

    def grid_search_svm(self, X_train_red_onehot, y_train_red):···


    def to_use_SVM(self, X_train_red_onehot, y_train_red, kf):···

    def to_test_SVM(self, X_test_red_onehot, y_test_red, X_train_red_onehot, y_train_red):···
```

Figure 23: Overview of our methods

The Python class "SVM_Model" contains the necessary set of Python methods for the implementation of the Support Vector Machine algorithm for Phishing websites detection. Figure 23 presents an overview of these methods. In the following I describe briefly the role of every method:

- **plot_cm():** is used for plotting the confusion matrix of the classifier**.**
- **grid_search_svm():** in this method, I use the Python method GridSearchCV() to estimate the best parameters for the SVM model on the Phishing URL dataset. This method returns a set of optimal parameters that are used in the subsequent steps of training and testing the SVM model.
- **to_use_SVM():** once the best estimator model is extracted, here, the best parameters are used for training the predictive model. In this step, I fit the training data to the SVM algorithm with its best setup parameters and calculate the accuracy metrics. Figure 24 shows the piece of Python code responsible on fitting the SVM model to the phishing websites training data. Moreover, Figure 25 shows the code required for the calculation of the accuracy measures.

```
svm_clf = svm_clf.fit(X_train_red_onehot[train], y_train_red[train])
```

Figure 24: Fitting model to the training data

```
accuracy = (tn+tp)/(fp+fn+tp+tn)
precision = tp/(tp+fp)
recall = tp/(tp+fn)
fpr = fp/(fp+tn)
f1 = 2*precision*recall/(precision + recall)
```

Figure 25: code required for the calculation of the accuracy measures

- **to_test_SVM():** this method provides predicting phishing websites test data with the SVM trained model, as well as saving the model for further usage. The local storage of the learned SVM model on our training data is done as shown in Figure 26 using the Python method "open()". Here, "svm_clf" is the trained model and "SVM_Final_Model" is the given file name of the trained model.

```
pickle.dump(svm_clf, open("SVM_Final_Model", 'wb'))
```

Figure 26: local storage of our training data

5.2 Execution Results

The estimation of the SVM best setup tells that a linear SVM is not enough for the prediction. The best SVM version for the studied Phishing websites data is rather a kernel SVM using a Radial Basis Function (rbf) with parameters "C" and "gamma". The parameter "gamma" defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The parameter "C" trades off correct classification of training examples against maximization of the decision function's margin.

```
Best parameters for the SVM Model applied on the Phishing websites data

{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
```

Figure 27: best parameters

Next, the SVM classification metrics are shown. The values presented in Figure 28 are the: accuracy, precision, recall, false positive rate, and the f1 score, respectively. It is to notice that the accuracy (96.11 %) and precision (96.23%) of the SVM model are relatively high. The SVM performs very well in the binary classification of the Phishing websites data.

```
SVM classifier metrics

[0.96110305 0.96238896 0.96788512 0.04738314 0.96512393]
```

Figure 28: classification metrics

As a next step, I applied the learned SVM model on the test data to predict the classification of the Phishing websites data. This step is done without the usage of the data classification labels. The accuracy of the learned model is shown in the following figure.

```
SVM classifier Test Accuracy

0.9629127091813658
```

Figure 29: the highest accuracy

As shown in Figure 29, the learned SVM model reaches 96.29% of higher accuracy in predicting the class of the phishing websites.

Notice: the same steps are repeated with other models just like with SVM model, the results of the implementation are shown in **page 79 Figure 46, 47, 48, 49, 50, 51.**

Table 5: comparison with similar works

Approach	Phishing sites number	Legitimate sites number	total	accuracy
Mohammad Nazmul Alam, Dhiman Sarma “random forest and decision tree”	1000	1000	2000	97%
Sonowal and Kuppusamy “PhiDMA”	667	995	1662	92,72%
Tan et al.” Vectors and Technical Approaches”	500	500	1000	91,51%
My approach	4898	6157	11055	97,06%

And here is a bar chart that shows the different accuracies taken from other approaches and mine.

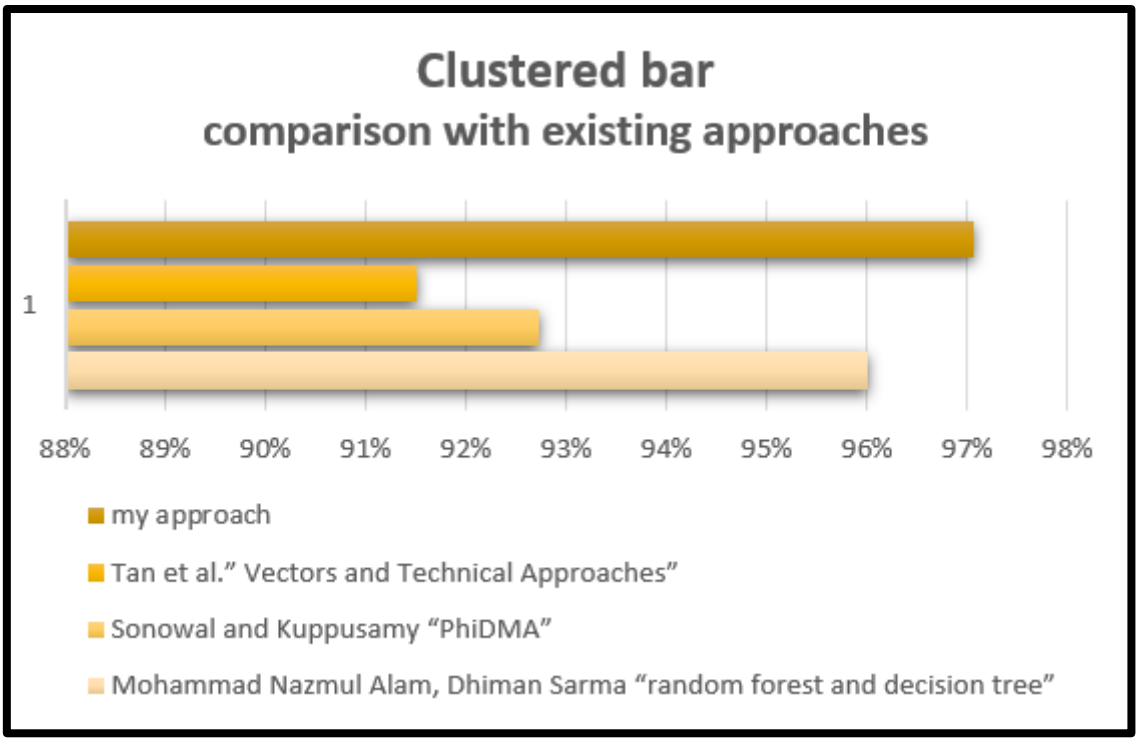


Figure 30: clustered bar - different approaches

-below a table showing accuracy results of the models

5.3 Convolutional neural network:

-After applying conv 1d to my dataset I got a very low accuracy and here’s the code and the compiling results in terminal.

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv1D, Conv2D, MaxPooling2D, MaxPooling1D

class CNN_Model:
    def create_cnn_model(self):···

    def train_cnn_model(self, X_train_red_onehot, y_train_red, X_test_red_onehot, y_test_red):     ···
        |

Epoch 1/3
277/277 [==============================] - 33s 10ms/step - loss: -48219.2369 - accuracy: 0.5519 - val_loss: -1574143.5000
- val_accuracy: 0.5568
Epoch 2/3
277/277 [==============================] - 2s 7ms/step - loss: -5564147.1736 - accuracy: 0.5513 - val_loss: -33522250.0000
 - val_accuracy: 0.5568
Epoch 3/3
277/277 [==============================] - 2s 7ms/step - loss: -59038478.1583 - accuracy: 0.5528 - val_loss: -178407952.00
00 - val_accuracy: 0.5568
```

Figure 31: CNN1D code and run output

-The figure 33 bellow shows the importations I need for my model and the figure 34 shows how we uploaded our dataset.

I am using the sequential API to be able to create the model Layer by Layer.

The dense layer is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer.

```
import pandas as pd
import numpy as np

from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import *

from keras import callbacks

from keras.layers import Dense, Dropout, Flatten, Conv1D, MaxPooling1D
from keras.layers.pooling import GlobalAveragePooling1D
```

Figure 32: import section

```
data = pd.read_csv('Dataset/Phishing.csv')
```

Figure 33: Loading the main dataset

-The Figure below shows the tuple **shape** that gives dimensions of the array…

```
>>> data.shape
(11055, 31)
```

Figure 34: Print data shape

-As we see, we have 30 features and 11055 datapoints.

```
>>> data.head(5)
   having_IP_Address  URL_Length  Shortining_Service  ...  Links_pointing_to_page  Statistical_report  Result
0                 -1           1                   1  ...                       1                  -1      -1
1                  1           1                   1  ...                       1                   1      -1
2                  1           0                   1  ...                       0                  -1      -1
3                  1           0                   1  ...                      -1                   1      -1
4                  1           0                  -1  ...                       1                   1       1

[5 rows x 31 columns]
```

Figure 35: Print first five rows

-the Figure above I used the head() function to show the first 5 rows of our dataset.

-the instruction below helps us to reflect our dataframe over its main diagonal by writing rows as colums and that makes it easier to read when display. The property T is an accessor to the method transpose().

```
>>> data.head(5).T
                              0  1  2  3  4
having_IP_Address            -1  1  1  1  1
URL_Length                    1  1  0  0  0
Shortining_Service            1  1  1  1 -1
having_At_Symbol              1  1  1  1  1
double_slash_redirecting     -1  1  1  1  1
Prefix_Suffix                -1 -1 -1 -1 -1
having_Sub_Domain            -1  0 -1 -1  1
SSLfinal_State               -1  1 -1 -1  1
Domain_registeration_length  -1 -1 -1  1 -1
Favicon                       1  1  1  1  1
port                          1  1  1  1  1
HTTPS_token                  -1 -1 -1 -1  1
Request_URL                   1  1  1 -1  1
URL_of_Anchor                -1  0  0  0  0
Links_in_tags                 1 -1 -1  0  0
SFH                          -1 -1 -1 -1 -1
Submitting_to_email          -1  1 -1  1  1
Abnormal_URL                 -1  1 -1  1  1
Redirect                      0  0  0  0  0
on_mouseover                  1  1  1  1 -1
RightClick                    1  1  1  1  1
popUpWidnow                   1  1  1  1 -1
Iframe                        1  1  1  1  1
age_of_domain                -1 -1  1 -1 -1
DNSRecord                    -1 -1 -1 -1 -1
web_traffic                  -1  0  1  1  0
Page_Rank                    -1 -1 -1 -1 -1
Google_Index                  1  1  1  1  1
Links_pointing_to_page        1  1  0 -1  1
Statistical_report           -1  1 -1  1  1
Result                       -1 -1 -1 -1  1
```

Figure 36: print head transposed

Below, columns attribute shows return the column labels of our dataframe.

```
>>> data.columns
Index(['having_IP_Address', 'URL_Length', 'Shortining_Service',
       'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix',
       'having_Sub_Domain', 'SSLfinal_State', 'Domain_registeration_length',
       'Favicon', 'port', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor',
       'Links_in_tags', 'SFH', 'Submitting_to_email', 'Abnormal_URL',
       'Redirect', 'on_mouseover', 'RightClick', 'popUpWidnow', 'Iframe',
       'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
       'Google_Index', 'Links_pointing_to_page', 'Statistical_report',
       'Result'],
      dtype='object')
```

Figure 37: Printing data columns

Below the sub-class counter allows us to count the different values in the attribute result.

```
>>> from collections import Counter
>>>
>>> classes = Counter(data['Result'].values)
>>> classes.most_common()
[(1, 6157), (-1, 4898)]
```

Figure 38: counting the unique values of classes

And that shows number of phishing and legitimate URLs.

Above shown Pandas describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame.

```
>>> data.describe().T
                             count      mean       std  min  25%  50%  75%  max
having_IP_Address          11055.0  0.313795  0.949534 -1.0 -1.0  1.0  1.0  1.0
URL_Length                 11055.0 -0.633198  0.766095 -1.0 -1.0 -1.0 -1.0  1.0
Shortining_Service         11055.0  0.738761  0.673998 -1.0  1.0  1.0  1.0  1.0
having_At_Symbol           11055.0  0.700588  0.713598 -1.0  1.0  1.0  1.0  1.0
double_slash_redirecting   11055.0  0.741474  0.671011 -1.0  1.0  1.0  1.0  1.0
Prefix_Suffix              11055.0 -0.734962  0.678139 -1.0 -1.0 -1.0 -1.0  1.0
having_Sub_Domain          11055.0  0.063953  0.817518 -1.0 -1.0  0.0  1.0  1.0
SSLfinal_State             11055.0  0.250927  0.911892 -1.0 -1.0  1.0  1.0  1.0
Domain_registeration_length 11055.0 -0.336771  0.941629 -1.0 -1.0 -1.0  1.0  1.0
Favicon                    11055.0  0.628584  0.777777 -1.0  1.0  1.0  1.0  1.0
port                       11055.0  0.728268  0.685324 -1.0  1.0  1.0  1.0  1.0
HTTPS_token                11055.0  0.675079  0.737779 -1.0  1.0  1.0  1.0  1.0
Request_URL                11055.0  0.186793  0.982444 -1.0 -1.0  1.0  1.0  1.0
URL_of_Anchor              11055.0 -0.076526  0.715138 -1.0 -1.0  0.0  0.0  1.0
Links_in_tags              11055.0 -0.118137  0.763973 -1.0 -1.0  0.0  0.0  1.0
SFH                        11055.0 -0.595749  0.759143 -1.0 -1.0 -1.0 -1.0  1.0
Submitting_to_email        11055.0  0.635640  0.772021 -1.0  1.0  1.0  1.0  1.0
Abnormal_URL               11055.0  0.705292  0.708949 -1.0  1.0  1.0  1.0  1.0
Redirect                   11055.0  0.115694  0.319872  0.0  0.0  0.0  0.0  1.0
on_mouseover               11055.0  0.762099  0.647490 -1.0  1.0  1.0  1.0  1.0
RightClick                 11055.0  0.913885  0.405991 -1.0  1.0  1.0  1.0  1.0
popUpWidnow                11055.0  0.613388  0.789818 -1.0  1.0  1.0  1.0  1.0
Iframe                     11055.0  0.816915  0.576784 -1.0  1.0  1.0  1.0  1.0
age_of_domain              11055.0  0.061239  0.998168 -1.0 -1.0  1.0  1.0  1.0
DNSRecord                  11055.0  0.377114  0.926209 -1.0 -1.0  1.0  1.0  1.0
web_traffic                11055.0  0.287291  0.827733 -1.0  0.0  1.0  1.0  1.0
Page_Rank                  11055.0 -0.483673  0.875289 -1.0 -1.0 -1.0  1.0  1.0
Google_Index               11055.0  0.721574  0.692369 -1.0  1.0  1.0  1.0  1.0
Links_pointing_to_page     11055.0  0.344007  0.569944 -1.0  0.0  0.0  1.0  1.0
Statistical_report         11055.0  0.719584  0.694437 -1.0  1.0  1.0  1.0  1.0
Result                     11055.0  0.113885  0.993539 -1.0 -1.0  1.0  1.0  1.0
```

Figure 39: data description statistics

-The info() function is used to print a concise summary of a DataFrame. This method prints information about a DataFrame including the index dtype and column dtypes, non-null values.

```
>>> data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11055 entries, 0 to 11054
Data columns (total 31 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   having_IP_Address            11055 non-null  int64
 1   URL_Length                   11055 non-null  int64
 2   Shortining_Service           11055 non-null  int64
 3   having_At_Symbol             11055 non-null  int64
 4   double_slash_redirecting     11055 non-null  int64
 5   Prefix_Suffix                11055 non-null  int64
 6   having_Sub_Domain            11055 non-null  int64
 7   SSLfinal_State               11055 non-null  int64
 8   Domain_registeration_length  11055 non-null  int64
 9   Favicon                      11055 non-null  int64
 10  port                         11055 non-null  int64
 11  HTTPS_token                  11055 non-null  int64
 12  Request_URL                  11055 non-null  int64
 13  URL_of_Anchor                11055 non-null  int64
 14  Links_in_tags                11055 non-null  int64
 15  SFH                          11055 non-null  int64
 16  Submitting_to_email          11055 non-null  int64
 17  Abnormal_URL                 11055 non-null  int64
 18  Redirect                     11055 non-null  int64
 19  on_mouseover                 11055 non-null  int64
 20  RightClick                   11055 non-null  int64
 21  popUpWidnow                  11055 non-null  int64
 22  Iframe                       11055 non-null  int64
 23  age_of_domain                11055 non-null  int64
 24  DNSRecord                    11055 non-null  int64
 25  web_traffic                  11055 non-null  int64
 26  Page_Rank                    11055 non-null  int64
 27  Google_Index                 11055 non-null  int64
 28  Links_pointing_to_page       11055 non-null  int64
 29  Statistical_report           11055 non-null  int64
 30  Result                       11055 non-null  int64
dtypes: int64(31)
memory usage: 2.6 MB
```

Figure 40: data info()

Splitting the dataset into 20% testing and 80% training

```
from sklearn.model_selection import train_test_split

X = data.iloc[:,0:30].values.astype(int)
y = data.iloc[:,30].values.astype(int)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=np.random.seed(7))
```

Figure 41: split dataset

```
>>> class_dist = pd.DataFrame(classes.most_common(), columns=['Class', 'Num_Observations'])
>>> class_dist
   Class  Num_Observations
0      1              6157
1     -1              4898
```

Figure 42: dataframe unique classes

Mapping a range of values to another can be easy by using the map() function.

```
data['Result'] = data['Result'].map({-1:0, 1:1})
```

Figure 43: mapping values

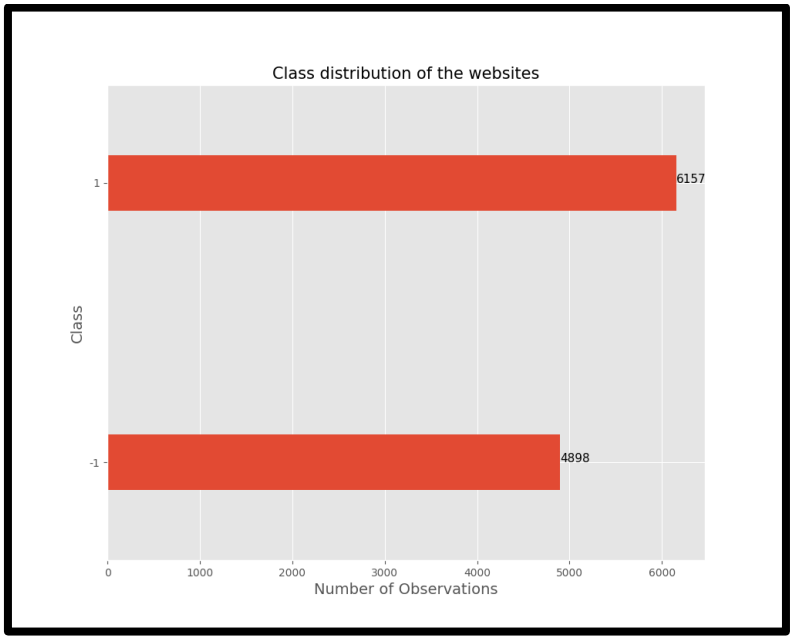


Figure 44: number of phishing/legitimate websites

And we got a high accuracy of 97 %.

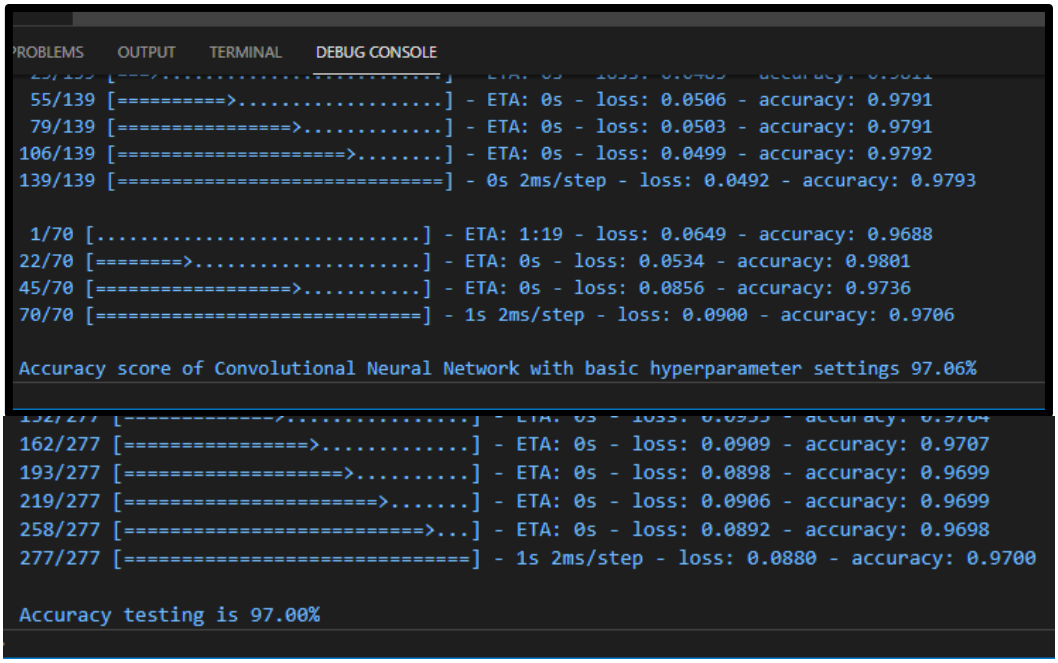


Figure 45: accuracy of testing and learning CNN final model

6 Comparative study

In this section, we perform a comparative study on the Phishing websites binary classification task. We investigate the performance of several potential machine learning algorithms in order to deduce which of these has the best classification achievement.

For this reason, I have implemented the following machine learning algorithms in object-oriented Python. The algorithms that I have taken into account are the following: Generalized Naive Bayes, K-Nearest Neighbors, XGBoost, Decision Tree, Random Forest, Logistic Regression, and the SVM algorithm.

For every algorithm, a Python class is created and called from the main python execution file "main_hamaidi.py"

In the following, the results of the execution of the considered algorithms for comparison are shown.

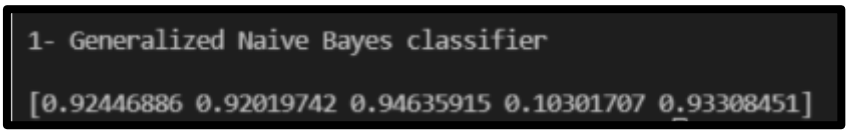


Figure 46: the classification results of GNB model

Figure 47: XGboost classification metrics

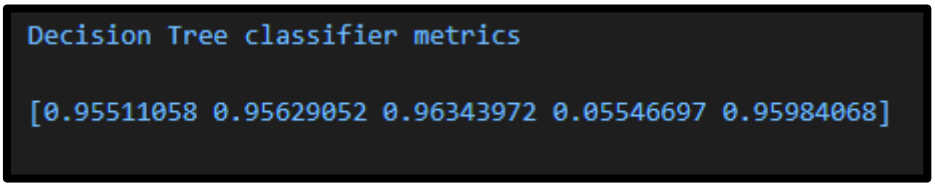


Figure 48: DT classification metrics

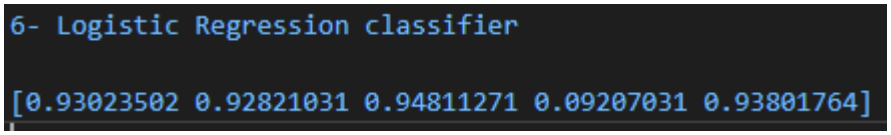


Figure 49: LR classification metrics

```
K-Nearest Neighbours classifier metrics

[0.94742262 0.9477164  0.95827329 0.06629807 0.95295169]
```

Figure 50: KNN classification metrics

```
Random Forest classifier metrics

[0.96415568 0.96443974 0.97136746 0.04481839 0.9678794 ]
```

Figure 51: Random Forest classification metrics

-My CNN model has the highest accuracy, so I chose it to deploy this model.

Table 6: Models accuracy comparative table

MODEL	ACCURACY
Random Forest	96%
SVM	96,11%
XGBOOST	93,32%
LOGISTIC REGESSION	93,02%
DECISION TREE	95,51%
GENERALIZED NAIVE BAYES	92%
KNN	95%
CONVOLUTIONAL NEURAL NETWORK	97,06%

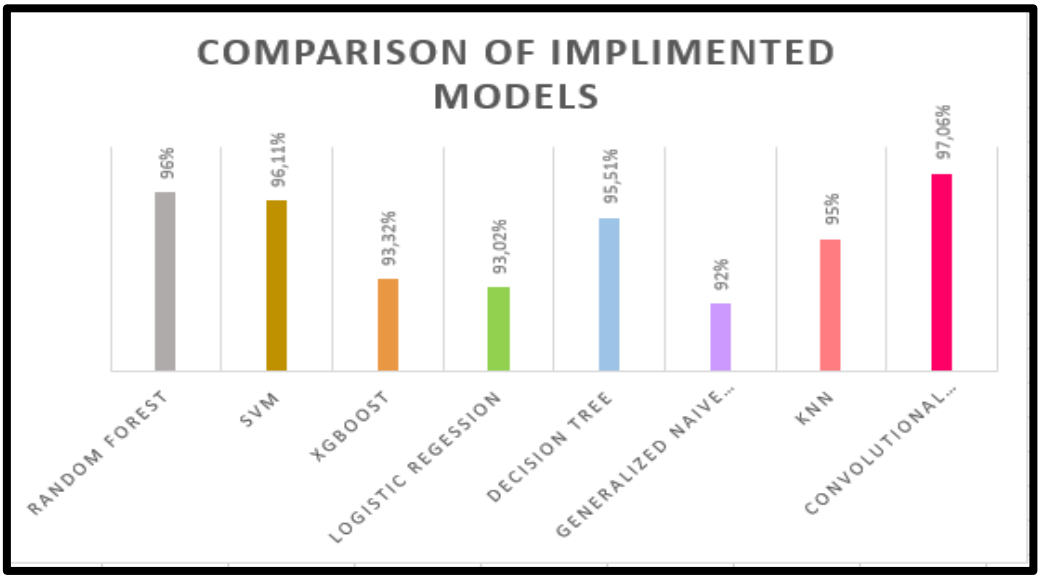


Figure 52: clustered bar - different accuracies

7 Running the interface

We can run the code only by using WSL version 2 and here is how to check or to switch to wsl2 version by running Windows PowerShell as administrator.

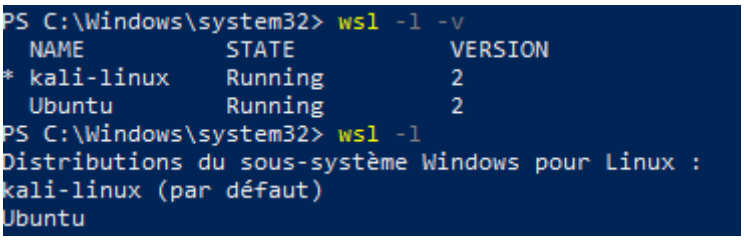


Figure 53: Checking my WSL version

If your Linux VM is running version 1, you can use the command WSL --set version Ubuntu

And thus, is after checking the build it should be greater than 16237.0

-To check the build:

Open PowerShell (as administrator) and type:

systeminfo | Select-String "^OS Name","^OS Version"

-Enable WSL in Windows >Go to turn on windows features >Enable Windows Subsystem for Linux

After that Open PowerShell in administrator mode and type:

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

-Go to Microsoft store and install your Linux distribution of choice.

-after lauching ubuntu install zsh.

sudo apt-get install zsh

-zsh will ask you to do some configuration. We will not do them now but during the installation of oh-my-zsh.

-to install oh-my-zsh

Before all we need to have git installed:

sudo apt-get install git

Then, use curl to install oh-my-zsh:

sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"

This will clone the repo and replace the existing ~/.zshrc with a template from oh-my-zsh.

-And this is how we run our project

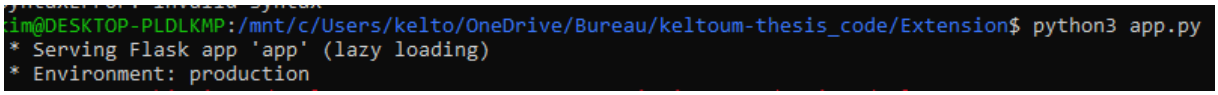


Figure 54: Running my project by using UBUNTU

8 Design

'Phishing Detector' is a web platform that authorized an URL as an input and predicted its status based on the CNN model as a Malicious URL or a benign one, and it has been integrated by using flask.

An input URL is expected to be complete. I used so many tools to extract different features during the running. I used curl and BeautifulSoup to scrape the webpage and its headers, tldextract to obtain a domain name from a given URL, string parsing to identify features extractable from the URL's text itself, WHOIS database to verify the authenticity of the link.

Except that, I found PageRank of the URL from Open Page Rank, page popularity using Alexa database, and the perceived reputation of the website using PhishTank API. After extracting these features, I re-encoded them using the trained OneHotEncoder and predict their legitimacy using the trained SVM. Finally, these results are served back to the user on the web platform.

These results are served back to the user on the web platform. The screens associated with the platform are shown in Figure 55, Figure 56 and Figure 57.

-the picture below shows the home page of my web platform.

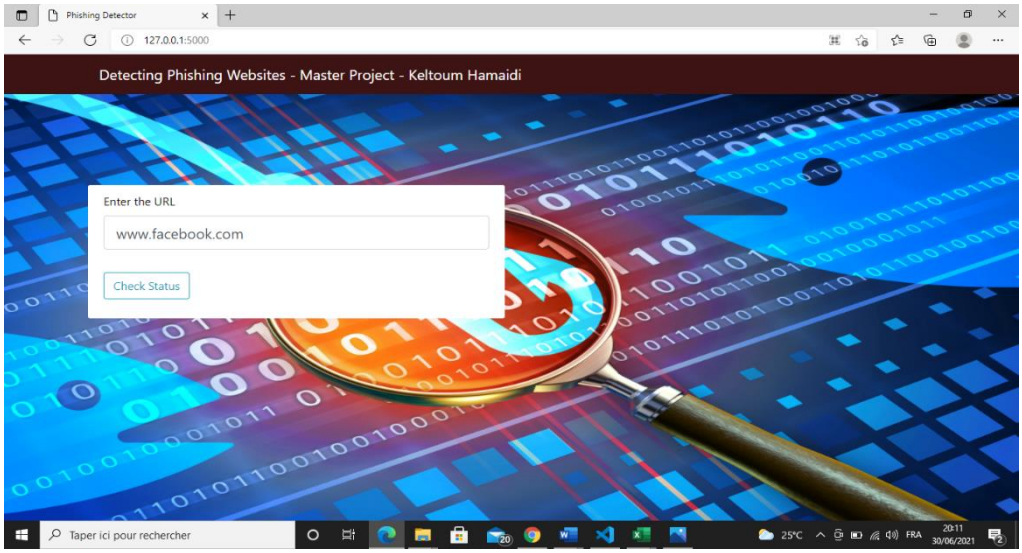


Figure 55: Home Page of my website

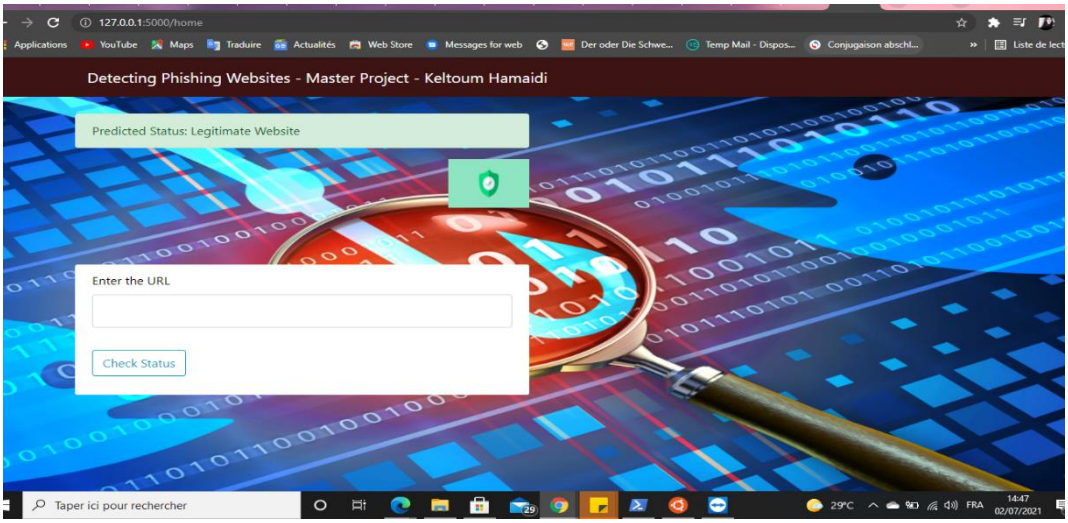


Figure 56: Prediction of a legitimate website

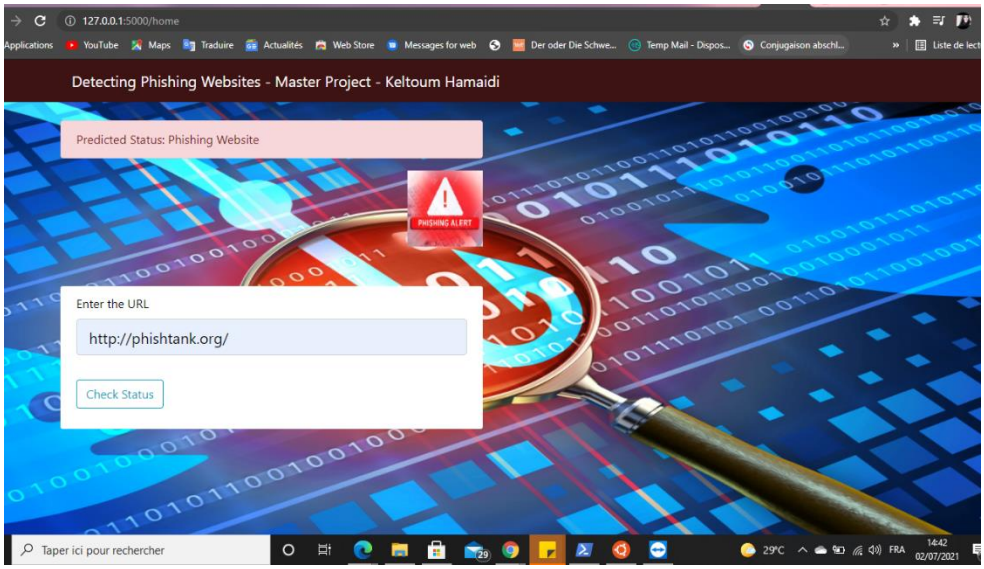


Figure 57: Prediction of a Phishing website

9 Conclusion

In this Chapter, I showed how the models work and compared my current CNN model with similar works based on the accuracy percentage and also, compared it with the results of the models I have implemented and picked one of them which has the highest accuracy, I have also shown a quick tutorial on how to enable and install the Windows Subsystem for Linux on a windows 10 pro machine. I have also explained each step of the algorithms I implemented and how my platform really works. And finally showed an example of detection on my interface.

General Conclusion

Phishing is an electronic attack that typically attacks computer networks. It first appeared in the 1990s. Today, phishing is a very well-known attack by hackers that aims to leave devastating consequences on economics, finance institutions, banking services. And that directly affects naïve users lives. More than 60,000 phishing websites were announced in March 2020. 96% of all attacks are expected to be for intelligence-gathering. 71% of all victims of the practice of extorting money or sexual favors from someone by threatening are younger than 18 years of age. Brand accounts of people who pretend to be another person for 81% of all spear-phishing attacks

This is why the detection of phishing sites has become a significant concern for researchers, several phishing detection approaches have been integrated into the solution, such as whitelists, blacklists, **machine learning**, visual similarity, and heuristics.

I presented a machine learning based approach combining several features to attain the best results. Among various classification models. After analyzing, I used a Convolutional neural network, which achieved an accuracy of 97.06% on training data with a set of 30 features spanning over broad categories of URL parsing, pages' source code and URLs embedded therein and domain-based statistics. I have then created a web-platform Phishing detector to deploy the obtained results.

I have used features from many domain crossing from URL to HTML tags, even d databases like WHOIS, Alexa, Pagerank. to check the traffic and status of the website.I was able to obtain a training accuracy of more than 97% and a testing accuracy of 97% as well, proving the effectiveness of the machine learning based technique to attack the problem of phishing websites. I provided the output as a user-friendly web platform which can further be converted to an extension or plugin to provide safe and healthy online space to the users.

Future work

Browser Extension: The platform can be converted into a browser extension. Phishing websites usually inflict losses to the users by acting as clickbaits. So, a browser extension can help prevent accidental land ups on these websites, by checking every URL which the browser tries to open, before actually allowing the user to land up on the page.

Caching Results in Database: Currently, for all queries, API hits and web scrapping is done every time a URL is provided as input. However, if some of these features and results can be cached in a database, the queries can be performed much more quickly and efficiently. So, if a seen URL is provided as input again, then the whole process of feature extraction and

classification can be skipped, and results can be served by looking up the database. But, as the websites are prone to updates, the results can be cached only for a few days, say a fortnight, after which the entry becomes invalid and the whole process has to be followed for it again.

Parallel Feature Extraction: As of now, features for an input URL are extracted sequentially. However, if the computation power is available, then most of the features can be extracted in parallel. The scrapping using BeautifulSoup and curl, WHOIS, Alexa, and other database lookups and string parsing can be done parallelly with work-stealing, thus bringing down the query time and increasing efficiency in the real world scenario.

References

I. Webography

[1] Anti-Phishing Working Group. (2018). Phishing Activity Trends Report. 2nd Quarter 2018.Retrieved from http://docs.apwg.org/reports/apwg_trends_ report_q2_2018.pdf.

[2] H. Bleau.(2017). Global Fraud and Cybercrime Forecast. https://www.rsa.com/en-us/blog/2016-12/ 2017-global-fraud-cybercrime-forecast.

[3] DhwaniMeharchandani.(December7, 2020),Staggering Phishing Statistics in 2020, Retrieved from https://securityboulevard.com/2020/12/staggering-phishing-statistics-in-2020.

[4] L'institut interrégional de rechercher des nations unies sur IA and criminalité et la justice.

[5] M. Chawki. «ESSAI SUR LA NATION DE CYBERCRIMINALITE.».

[6] Kaspersky, «What is Cybercrime?»

[7] «Définition d'arnaque (scam),» [En ligne]. Available: softwarelab.org/fr/arnaque/

[8] Was ist Phishing? Die Definition und 5 Hauptarten. https://softwarelab.org/de/was-ist-phishing/

[9] Preethi. (05-09-2018) 14 Types of Phishing Attacks That IT Administrators Should Watch For.blog.syscloud.com/types-of-phishing/

[10] F. T. M. Rami M. Mohammad. Phishing Websites Features.

[11] «Phishing URL Detection with ML,» Towards Data Science

[12] D. W. AP Felt. (2011). Phishing on mobile devices.

[13] Supervised vs Unsupervised vs Reinforcement,Posted.(January 29, 2020). Surbhi Arorain Machine Learning, aitude.com

[14] L.LERMA. Les systèmes de détection d'intrusion basés sur du machine learnin. thèse de doctorat, Université libre de Bruxelles. .

[15] S. Zhao, Z. Xu, L. Liu, M. (2017).GuoTowards Accurate Deceptive Opinion Spam Detection Based on Word Order-Preserving CNN. pp. 1-8. pdfs.semanticscholar.org/1687/0bed28831f6bd49a0228177351d1870fafd1.pdf

[16] H. Yuan, X. Chen, Y. Li, Z. Yang, and W. Liu, 'Detecting Phishing Websites and TargetsBased on URLs and Webpage Links', in 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 3669–3674, doi: 10.1109/ICPR.2018.8546262.

[17] W. Ali and A. A. Ahmed, 'Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting', IET Information Security, vol. 13, no. 6, pp. 659–669, 2019, doi: 10.1049/iet-ifs.2019.0006.

[18] A. Correa Bahnsen, E. Contreras Bohorquez, S. Villegas, J. Vargas, and F. A. Gonz´alez. (April 2017). Classifying phishing URLs using recurrent neural networks. in Proceedings of APWG Symposium on Electronic Crime Research (eCrime), pp. 1–8, IEEE, Scottsdale, AZ, USA.

[19] L. Hung, Q. Pham, D. Sahoo, and S. C. H. Hoi. 2018. Urlnet: learning a URL representation with deep learning for malicious URL detection. arxiv.org/abs/ 1802.03162.

[20] E. Buber, B. Dırı, and O. K. Sahingoz. Detecting phishing attacks from url by using nlp techniques. in 2017 International Conference on Computer Science and Engineering (UBMK), Oct 2017.

[21] W. Wang, F. Zhang, X. Luo, S. Zhang: PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks, Security and Communication Networks. Volume 2019. doi.org/10.1155/2019/2595794

[22] Y.Fang , C.Zhang ,C.Huang ,L.Liu, and Y.Yang"Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism",April 29, 2019,Digital Object Identifier 10.1109,ACCESS.2019.2913705

[23] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, ''An empirical analysis of phishing blacklists,'' in Proc. 6th Conf. Email Anti-Spam (CEAS), Sacramento, CA, USA, 2009, pp. 1–10.

[24] A. Vazhayil, N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, ''PED-ML: Phishing email detection using classical machine learning techniques,'' in Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal. (IWSPA), A. D. R. Verma, Ed. Tempe, AZ, USA, 2018, pp. 1–8

[25]M. Nguyen, T. Nguyen, and T. H. Nguyen. (2018). ''A deep learning model with hierarchical LSTMs and supervised attention for anti-phishing.''[Online]. Available: arxiv.org/abs/1805.01554

[26] Anti-Phishing Working Group. (2016). Phishing Activity Trends Report 4th Quarter 2016. [Online]. Available: docs.apwg.org/ reports/apwg_trends_report_q4_2016.pdf

[27] Anti-Phishing Working Group. (2015). Phishing Activity Trends Report 1st-3rd Quarter 2015. [Online]. Available: docs.apwg.org/ Preports/apwg_trends_report_q1-q3_2015.pdf

[28] L. M. Form, K. L. Chiew, S. N. Sze, and W. K. Tiong, ''Phishing email detection technique by using hybrid features,'' in Proc. 9th Int. Conf. IT Asia (CITA), Aug. 2015, pp. 1–5.

[29] Microsoft. (2018). Microsoft Security Intelligence Report. [Online]. Available: clouddamcdnprodep.azureedge.net/gdc/gdcVAOQd7/origin

[31] Ai-united,Der Random Forest (Zufallswald) Algorithmus

[33] N. R. Gade, U. G J Reddy,February 2014,"A Study Of Cyber Security Challenges And Its Emerging Trends On Latest Technologies"[online].Available: researchgate.net/publication/260126665_A_Study_Of_Cyber_Security_Challenges_And_Its_Emerging_Trends_On_Latest_Technologies

[35]Nabi, J. (2019, May 24). Machine Learning —Fundamentals - Towards Data Science. Medium. https://towardsdatascience.com/machine-learning-basics-part-1-a36d38c7916

[37] Seif, G. (2021, February 14). A Guide to Decision Trees for Machine Learning and Data Science. Medium. https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956

[38]Luber, S. (2019b, March 19). Was ist Keras? BigData-Insider. https://www.bigdata-insider.de/was-ist-keras-a-726546/

[39]Luber, S. (2019b, March 19). Was ist TensorFlow? BigData-Insider. https://www.bigdata-insider.de/was-ist-tensorflow-a-684272/

[40] A.Mishra.(2020). XGBoost an efficient implementation of gradient boosting.https://www.datascience.foundation/datatalk/xgboost-an-efficient-implementation-of-gradient-boosting.

[41]Ray, S. (2020, December 23). Commonly used Machine Learning Algorithms (with Python and R Codes). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

[42]Donges, N. (2021, June 2). A complete guide to the random forest algorithm. Built In. https://builtin.com/data-science/random-forest-algorithm

[43] Brownlee, J. (2020, August 15). Linear Regression for Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/linear-regression-for-machine-learning/

[48] "Weka 3: Machine Learning Software in Java", Available: cs.waikato.ac.nz/ml/weka/

[49]Luber, S. (2019, March 19). Was ist eine Whitelist und Blacklist? Security-Insider. https://www.security-insider.de/was-ist-eine-whitelist-und-blacklist-a-667574/

[50] A. Kumar, J. Kuri.(2004).Heuristic-Based Algorithm.https://www.sciencedirect.com/topics/computer-science/heuristic-based-algorithm.

[51] A Personalized Whitelist Approach for Phishing Webpage Detection. (2012, August 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/6329190

[52] M. a. P. K. A. (December 2016). Gaurav Varshney. A survey and classification of Web.

[53] A. N. M. a. j. H. Mohuiddin Ahmed, «A survey of network anomaly detection techniques».

[54] S. Jagadessan. «URL Phishing Analysis using Random forest.».

[55]Malicious web content detection using machine leaning. (2017, May 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8256834

[56] R. M. Mohammad, F. Thabtah, and L. McCluskey. (2012). An assessment of features related to phishing websites using an automated technique. In 2012 International Conference for Internet Technology and Secured Transactions, pages 492–497

[57] W. (n.d.). Was istkünstliche Intelligenz? WFB. Retrieved June 13, 2021, from https://www.wfb-bremen.de/de/page/stories/digitalisierung-industrie40/was-ist-kuenstliche-intelligenz-definition-ki

II. Bibliography

[30] Zaccone, G., & Karim, M. R. (2018). Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python, 2nd Edition (2nd Revised edition). Packt Publishing.

[32]Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). Python Deep Learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow, 2nd Edition. Packt Publishing.

[36]: Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques (The Morgan Kaufmann Series in Data Management Systems) (3rd ed.). Morgan Kaufmann.

[34]: Numerisches Python – Arbeiten mit NumPy, Matplotlib und Pandas [online] available: tinyurl.com/p78wxr9u