PEOPLES DEMOCRATIC REBUPLIC OF ALGERIA

MMINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCHE

UNIVERSITY MOHAMED BOUDIAF - M'SILA

FACULTY: MATHIMATICS AND COMPUTER SCIENCE

DEPARTEMENT: COMPUTER SCIENCE

N°: ……..

DOMAIN: MATHIMATICS AND COMPUTER SCIENCE

BRANCH: COMPUTER SCIENCE

OPTION: RTIC

Dissertation submitted to obtain Master degree

By:

- **BELOUADAH KHALIL**
- **SALEM MOHAMED EL-AMINE**

SUBJECT

WEB BROWSER EXTENSION FOR DETECTING COVID-19 THEMED MALICIOUS WEB CONTENT

Publicly defended before the jury composed of:

Ms. SAOUDI LALIA	University of M'sila	Supervisor
Dr. Heragmi Kamel Eddine	University of M'sila	Chair
Dr. Oueldmehamdi Nadjib	University of M'sila	Examinetor

Academic year 2020/2021

Acknowledgment

At the beginning, we always express our gratitude to Almighty God in his conciliation in drawing up a road map for this humble work.

Idealism is what is needed, but unfortunately our work was not perfect due to lack of time for us before finishing this project. We would like to thank everyone who had a hand in the completion of the project, and we thank the professor and supervisor of the subject in particular SAOUDI LALIA for the mechanism that provided us with everything that is a brick in the construction of this project, and she had patience with us in the work of the project and corrected it repeatedly, and always Thanks and a thousand thanks, it was a pleasure to work with you even though we could not provide everything we had.

BELOUADAH KHALIL
SALEM MOHAMED ELAMINE

Table of content

List of figures and tables

List of figures

List of tables

List of equations

General Introduction

GENERAL INTRODUCTION

Context of the Study

While the pandemic has kept organizations closed due to social distancing, they are developing workers electronically in line with the new process of "working from home and keeping everyone safe". This has created opportunities for cyber-attacks to circumvent new news and upgraded processes to hack home computers.

These attackers were using websites with names associated with the COVID-19 names, which have been dubbed the COVID-19 malicious website. These sites often contain false information, fake forms, false payments, scams, or malicious payloads to steal sensitive information or infect computers. It meets the needs of all people in terms of communication and business, as well as entertainment, in addition to banking, shopping and many other activities that take place through web applications.

People became concerned about their information and data as a result of this threat, prompting them to seek out methods or tactics to protect it.

Statement of Problem

The problem of detecting malicious web content, URLs and phishing websites, have been studied in the literature prior to the COVID-19 pandemic , However, these studies do not consider malicious contents related to the COVID-19 pandemic, which are the new most used content for hacking.

COVID-19 themed malicious web sites are dramatically increased, this new orientation imply that it has become more challenging to distinguish between legitimate COVID-19 web content from malicious one.

Objective

In order to prevent malicious web content with the theme of COVID-19, we suggest a Google Chrome browser extension (CovProtectWeb) to detect known types of malicious content and news related to COVID-19, in order to provide a safe environment for web browsing and make them more comfortable.

For this aim, we use a machine learning algorithms with a set of developed COVID-19 malicious web features and collect a dataset of 400 websites between

benign and malicious site which contain the data of the most visited sites in the confinement period in order to compare it with the exploitation sites.

Motivation

During this COVID-19 time, the COVID-19 pandemic and the ensuing lockdown has accelerated the use of online technologies such as contactless digital payment [01].
Internet criminal activity spread. The spread of malicious content, especially the names associated with COVID-19, and the cyber-attacks of the most visited sites in the Internet insanely increased due to the value of information and financial transactions. According to a French bank, in April, total debt and direct transfers accounted for 45% of payments in France, which is a 12% increase compared to the previous months [02]. In the US, between February and April, cyber-attacks targeting the financial sector increased [03]. Global spending on cybersecurity will have to increase too.

Report outline

This document is divided into four chapters:

• **The First Chapter:** It focused on explaining the different types of malicious web content, the URL structure of malicious COVID-19 themed site, and the reasons why attackers exploited the topic of COVID-19.

• **The Second Chapter:** Focus on an overview of existing approaches for detecting harmful content carrying COVID-19 as a topic, closed with a comparison between them.

• **The Third Chapter:** Presenting our claimed approach equipped with our conceptual model and techniques to detect malicious web content bearing COVID-19 as a topic.

• **The Fourth Chapter:** It will contain the implementation and experimentation procedures that we will take to develop our approach with the results we have achieved.

• **And finally,** a general conclusion of the relevant topic.

CHAPTER 01

Malicious Websites Themed COVID-19 & URL Architecture

1.1. Introduction

The development of the Internet has resulted in an exponential increase in web content. Web-based scams, such as drive-by download attempts and phishing, have also increased year after year. URL blacklists are extensively used to prevent such attacks. URL blacklists, on the other hand, are insufficient because they are unable to detect newly created dangerous URLs.

This chapter presents malicious URL content types and COVID-19 Themed Malicious Websites.

1.2. What is a Malicious URL?

A malicious URL is a link that has been constructed with the intent of spreading frauds, attacks, and frauds. You can download ransomware, viruses, Trojans, or any other sort of malware by clicking on an infected URL, which could put your device or, in the case of a corporation, your entire network at danger [04].
A malicious URL can also be used to persuade you to divulge personal information on a phony website. It's worth noting that connections to malware aren't the only form of hazard that can propagate via the Internet; there are a variety of threats.

In this section, we've picked the most common types of malicious content:

1.2.1. Conditional redirects

When hackers get access to a website, they can include malicious code that causes certain users to be sent to another site. To construct these malicious redirects, attackers commonly use the **.htaccess** or **web.config** files to change web server configuration rules, add server-side scripts, or even inject client-side JavaScript [05].
In order to achieve the conditional elements, attackers limit redirection to referrers or user agents on a regular basis in order to target certain visitors and escape discovery.
The ultimate destinations are frequently tainted with malware or set up for phishing, while search engines blacklist the originating website.

1.2.2. Search Engine Optimization ‘SEO’ spam

Although search engines are constantly updated to detect and reject unscrupulous websites from their results, this does not always prevent criminal actors from hijacking reputable websites' rankings.

SEO Spam, also known as Spamdexing, is the activity of spamming search engines. SEO spam uses a variety of techniques to alter the relevance or prominence of search keywords indexed by search engines, such as link building and repeating unrelated phrases. Search engine spam is an attempt to manipulate search engine rankings in order to drive website visitors to a hacker-created fraud. To accomplish this, hackers get access to a normal, healthy website and then inject keywords and connections to another website they have created with the intent of defrauding consumers [06].

Spam content generally includes medicines, online gambling, pornography, and other less savory topics, and it allows an attacker to increase their own ranks with little effort.

The most frequent SEO hacks include injecting invisible links into the pages of an existing website. This essentially produces entire website parts, which can range in size from a few pages to thousands.

1.2.3. Malicious JavaScript

JavaScript is a widely used programming language for implementing complicated functionality on web pages, CMS platforms, and other large applications. The browser executes this sophisticated programming language, which exposes new and unique vulnerabilities – and is especially valuable for attackers once they have access to a compromised environment.

One of the most serious vulnerabilities to modern websites is cross-site scripting, sometimes known as XSS. Certain attacks inject scripts that target authorized site administrators; when run in the context and with authorization of these users, the malicious script might surreptitiously create new rogue admin users or weaken site security settings, making it easier to take over the entire site.

Attackers utilize JavaScript code to exploit vulnerabilities in the user's browser, browser plugins, or to deceive the victim into clicking on a link hosted by a malicious host. Drive-by-download is one of the most common assaults carried out with malicious JavaScript.

involving the installation (and execution) of malware on the victim's workstation Scripts that misuse system resources, such as opening windows that never close or creating a huge number of pop-up windows, are another type of JavaScript-based attack [07].

Many strategies for detecting malicious Java Scripts are available in the literature, but all of them have drawbacks. Some present detection systems rely on prior knowledge about malware; therefore they may be successful against well-known attacks but ineffective against zero-day threats [08].

Another problem of many malicious JavaScript code detectors is that they are designed to recognize specific sorts of attacks; hence, to avoid them, attackers typically mix up multiple attack types [09].

Despite the fact that JavaScript is only performed on the client side and not on the server, it may be used to interact with a website by doing background queries. Attackers can use these requests to gather information about a visitor's browser, engage in asynchronous operations, or even inject unwanted spam content into a web page without having to refresh it.

1.2.4. Defacements

Hackers employ a variety of techniques to deface websites. Hackers typically inject infected code into the site's script, allowing them to seize control of the website. They can acquire access to the website and any sensitive content using this control. Hackers may utilize a virtual private network (VPN) to remain anonymous when defacing a website.

Attackers can gain access to the website by exploiting a flaw in the program while using automated scanning software to detect website vulnerabilities. Hackers can impersonate authorized users in order to access remote files on websites and execute their own commands.

1.2.5. Phishing

Phishing is derived from the verb "to phish." In other words, the goal of phishing is to lure the victim into providing personal data and information.

Phishing attacks are when an attacker pretends to be someone, a company, or a government organization in order to lure and fool or trick someone through an:

a. **Email spoofing:** A hacker may send a message to someone posing as an authoritative person from a well-known website or a trusted partner institution, requesting information.

b. **Text message, phone call, called by direct social engineering:** Other attempts are more direct and particular, such as someone attempting to establish a discussion through instant chat.

c. **Social network or fake website 'fabrication'**: Someone could create a mockup of a website that looks like the login page for an actual software firm but has a slightly different URL. If a person is duped, he or she may input their login and password without thinking.

That evil actor impersonates someone else in order to obtain information, privileged access, or login credentials [10].

Phishing campaigns can take a variety of forms, including:

- Fake login sites for well-known firms,
- Online banking portals,
- Social media landing sites,
- Even webmail portals

Strange requests from what appear to be recognizable companies, user experiences with a sense of urgency to manipulate victims into skipping over details they might otherwise notice, or carefully crafted login pages that try to convince users they are logging into a valid service are all common characteristics of website phishing.

Phishing is difficult to detect because criminal actors hide their malicious pages deep inside the directory structure of a website.

1.2.6. Backdoors

When hackers acquire access to a website, they frequently install harmful code that allows them to keep or reclaim unlawful access beyond the original infection. Backdoors are one of the most prevalent types of malware detected on compromised websites.

Backdoors might be simple or exceedingly complicated. Remote code execution backdoor uploaders, for example, allow attackers to execute code through POST, GET, or COOKIE requests without the permission of the website. Other uploaders may allow hackers to upload dangerous files to a website's disk, such as spam or hack tools.

Modifying or creating new user accounts with elevated rights is another popular strategy.

Backdoors on webservers are exploited for a variety of malevolent purposes, including:

- Theft to a website
- Theft of data
- Hijacking a server
- DDoS (distributed denial of service) attempts
- Infecting visitors to a website (watering hole attacks)

To get around security regulations prohibiting the upload of files larger than a particular size, backdoor Trojan insertion is frequently done in two steps. The first step is to install a dropper, which is a tiny program whose primary purpose is to retrieve a larger file from a remote site. The second phase begins with the backdoor script being downloaded and installed on the server.

1.2.7. Bots/Botnets

A bot is a software application that performs automated tasks on command. They're used for legitimate purposes, such as indexing search engines, but when used for malicious purposes, they take the form of self-propagating malware that can connect back to a central server.

Usually, bots are used in large numbers to create a botnet, which is a network of bots used to launch broad remotely-controlled floods of attacks, such as DDoS attacks. Botnets can become quite expansive. For example, the Mirai IoT botnet ranged from 800,000 to 2.5M computers.

Botnet Example: Echobot is a variant of the well-known Mirai. Echobot attacks a wide range of IoT devices, exploiting over 50 different vulnerabilities, but it also includes exploits for Oracle WebLogic Server and VMWare's SD-Wan networking software. In addition, the malware looks for unpatched legacy systems. Echobot could be used by malicious actors to launch DDoS attacks, interrupt supply chains, steal sensitive supply chain information and conduct corporate sabotage.

1.3. COVID-19 Themed Malicious Websites

Using current events to drive nefarious activity is a popular approach utilized by hackers. With the development of the COVID-19 worldwide pandemic and the overwhelming thirst for the most up-to-date information, it might be difficult for users to guarantee that they click on trustworthy pages. Until recently, the world has witnessed harmful behaviour via nearly every online outlet: Email, social media, text and phone communications [11].

The current scale of malicious activity aiming to exploit COVID-19 differs throughout the world.
Threat actors are exploiting the emerging coronavirus (COVID-19) pandemic to register malicious websites disguised as COVID-19-related resources, such as news and public health updates, maps illustrating the spread of COVID-19, or even requests for donations to charitable campaigns and emergency funds. Malicious websites are used by threat actors to mislead the general population. [12]

Harmful websites that use COVID-19 as a theme, or COVID-19 themed malicious websites [12], are an emerging assault. Because home computers are frequently more vulnerable to attack than work computers, the implications of COVID19 themed fraudulent websites can be dramatically increased if firms adopt a "work from home" strategy.

1.3.1. Fake COVID-19/Coronavirus-related domains

During the epidemic of COVID-19 Websites cater to everyone's needs in terms of communication and business, as well as entertainment, in addition to banking services, shopping and many other activities that take place through web applications.

We observe websites use with names associated with the COVID-19 names, which the COVID-19 malicious website has dubbed. These sites often contain false information, fake forms, false payments, phishing scams, or malicious uploads to steal sensitive information or infect computers [13].
The COVID-19 pandemic has incurred many new cyber-attack vectors. Many of these cyber-attacks incorporate COVID-19 themed factors into phishing, malware, and

scamming schemes for various malicious goals (e.g., monetary benefits, stealing credentials, stealing credit card numbers, or identity theft). For example, there is reportedly a 148% increase in ransomware attacks in March 2020 compared with February 2020, where many attacks are initiated by malicious websites abusing victims' trust. [14]

1.3.2. Threats web on COVID-19

We look at some of the known Coronavirus scams going on right now

Due to quarantine restrictions affecting offices around the world, many components of daily work, from meetings to presentations and collaborative tasks, have been contacted via the Internet.

Over the following weeks and months, it is expected that hacking organizations and cybercriminals will continue to exploit the COVID-19 pandemic. Cybercriminals are using popular Internet tools, software sharing, and file attachments in their scams as they adapt to new operating methods:

- **Spam**: Sending malicious email in the early stages of COVID-19: There are now Business Email Hacking (BEC) schemes that use disease as bait. BEC schemes often work by tricking targets into transferring money to a criminal pretending to be someone from the same company. Email takes advantage of the current health situation to call for immediate action.
- To boost sales, some sellers include the term "coronavirus" in the title or text of their ads. They sell virus-themed merchandise or seek partners for collaborative projects.
- Some people are discussing how the infection can be exploited to trick social engineering. To avoid important transaction verification questions.
- Phishing using the subject of coronavirus or COVID-19 as a lure,
- malware distribution using coronavirus- or COVID-19-themed lures,
- Registration of new domain names containing wording related to coronavirus or COVID-19, have all been observed as threats.
- Attacks on remote access and teleworking infrastructure that has been implemented recently—and frequently quickly.

- Despite warnings about the risks of the Internet, hackers are using fixation with Coronavirus targeting with malware.
- Here's another disagreement on the issue, with the proviso that enticing "coronavirus maps" are now being used to infect victims' PCs with malware. Although warnings regarding the map location have already been given, warning users that such downloads will result in "the theft of credentials such as usernames, passwords, credit card numbers, and other sensitive information," laboratories are looking at this problem in particular.

1.3.3. Threat map with COVID-19

Malicious URLs include phishing-related sites and domains that remove malware (adware and ransomware).Where the hacker mistakenly provided people with access to harmful URLs containing COVID, COVID-19, coronavirus, or ncov. The number of times people attempted to access fraudulent COVID-19-related URLs peaked in April, with continued activity in May and June. However, activity picks up again in the third quarter, notably in August and September.

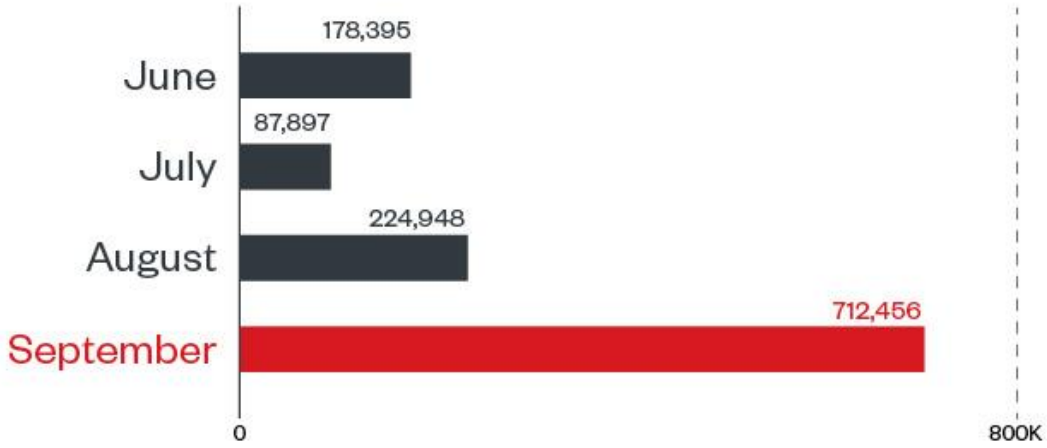


Figure1.1: Instances that malicious COVID-19 related URLs were accessed [15]

1.3.4. Risks and types of cyber-attacks on COVID-19 and defense tips

In this section we present how to avoid hackers during this crisis and how to protect you self.

The world is seeing an increase in certain types of attacks. Most of these measures can be aborted with proper electronic hygiene, including increased users' alertness. Or by not clicking on the malicious link, he made the following prominent frauds that some of our companies see, and we find that companies recommend employees to:

- **Phishing and Malspam** - These companies have reminded employees to be careful when opening emails about COVID-19, especially those coming from outside the organization. They should be careful when entering credentials into a website, linking them from an email, text message, or social media account, or when downloading attachments.
- **Credential Stuffing** — it may have been required to make services available to employees remotely since there was no time to safeguard accounts with multi-factor authentication (MFA). Along with safeguarding accounts using MFA, personnel should ensure that all passwords are safe and never reuse passwords across multiple accounts.
- **Ransomware** — in some situations, Malspam emails that initiate a ransomware attack may contain a COVID-19 lure. While avoiding ransomware attacks is the ideal outcome, being prepared with backups is the second best.
- **Remote Desktop Protocol (RDP) Targeting** - As the number of employees connecting remotely grows, so will the number of computers with open RDP (port 3389) that might be scanned. While your staffs require remote access to systems, limited and protected access via VPN can lower the attack surface.
- **Distributed Denial of Service (DDoS)** Attacks Stopping an assault is very dangerous with a remote workforce. Because more people are attempting to access the services at the same time, a bigger, remote workforce might operate as an unintentional DDoS assault. Increased ready-made bandwidth allowances are a superior alternative for dealing with these possibilities and ensuring protection against DDoS attacks. Disabling useless services temporarily to allow for additional bandwidth is a

- **Potential solution**, as is prohibiting workers from streaming films, music, or other streaming services over a VPN.

1.4. URL (Unified Resource Locator)

1.4.1. What is a URL?

A URL (Uniform Resource Locator) is a one-of-a-kind identifier that can be used to find a resource on the internet. A web address is another term for it. URLs are made up of several components, including a protocol and a domain name, that instruct a web browser how and where to find a resource.

End users access URLs by typing them into a browser's address bar or by clicking a hyperlink on a webpage [16].

1.4.2. How is a URL structured?

The URL includes both the name of the protocol used to access a resource and the name of the resource itself. The first portion of a URL specifies which protocol should be used as the main access method. A URL can also describe a path to a specific page or file within a domain. The second element indicates the IP address or domain name — and perhaps subdomain — where the resource is located. [17]

A URL usually looks something like this:

- It (usually, but not always) starts with "http://" or "https://"
- It is often followed by "www"
- The name of the website you want to visit
- then by specific directories where the information you want to read is stored, separated by / marks
- And finally, the location of the page you want to read.

Figure1.2: Basic URL structure [18].

A URL is composed of different parts, some mandatory and others optional. The most important parts are highlighted on the URL below:

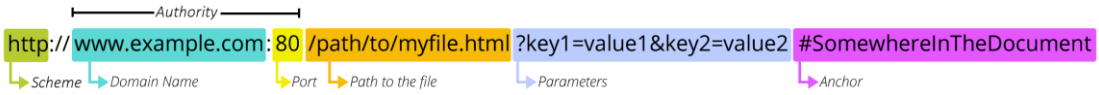


Figure1.3: highlight URL Structure [19].

1.4.2.1. Protocol

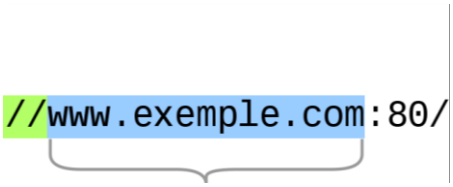
HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) + HTTP Strict Transport Security (HSTS) are three of the most often used URL protocols. The "Web" is built on these two protocols, however other protocols such as mailto: (which allows you to start an electronic mail client) and ftp: (which allows you to transfer files) are sometimes supported by browsers. So don't be shocked if you run into these additional protocols. FTP for file transfers, mail to for web resources, mail to for email addresses.

Protocol

Figure1.4: Structured URL: Protocol [17].

1.4.2.2. Domain name

It specifies the web server to which the browser sends content for exchange. You can use an IP address instead of a domain name, although this is less practical (and therefore less used on the Web).



Domain Name

Figure1.5: Structured URL: Domain name.

1.4.2.3. Port

This value denotes the technological "door" to utilize in order to gain access to server resources. Because the browser uses the normal ports associated with the Protocols, this element is usually missing (80 for HTTP, 443 for HTTPS). If the server's port isn't the default, you'll need to specify it

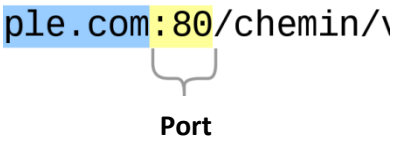


Figure1.6: Structured URL: port.

1.4.2.4. Path

This path typically related to a "physical" path on the server in the early days of the Web. This path is now merely an abstraction controlled by the web server; it no longer correlates to a "physical" reality.

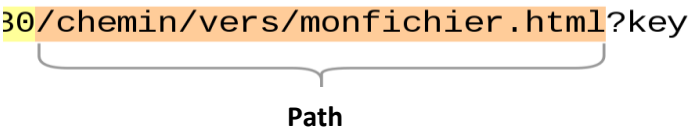


Figure1.7: Structured URL: path.

1.4.2.5. Parameters

These parameters are constructed as a list of key / value pairs, each element of which is separated by an ampersand (&). The web server can use these settings to perform additional actions before sending the resource. Each web server has its own rules for settings. In order to know them, the best is to ask the owner of the server.

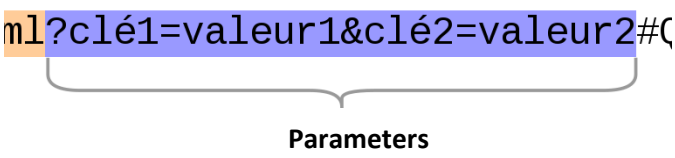


Figure1.8: Structured URL: parameters.

1.4.2.6. Fragment

This designates a given location in the resource. An anchor represents, in a way, a bookmark within the resource. Adding an anchor to a URL allows the browser to display the resource at the location of that bookmark. For an HTML document, for example, the browser will scroll the page down to the anchor level. For an audio or video document, the browser will go to the instant represented by the anchor. Note also that the part of the URL located after the # is never sent to the server with the request.

Figure1.9: Structured URL: Fragment.

Note: There are other fragments and other rules for URLs but these are not relevant for web development and are not necessary to be able to build fully functional URLs.

1.5. How to use URLs

Any URL can be entered in the address bar of the browser to access the corresponding resource, but that's not all!

1.5.1. Take advantage of URLs in HTML

HTML language - allows you to take advantage of URLs:

- By creating links to other documents using the <a> element;
- Link documents with associated resources using <link> and <script> elements;
- Displaying media like images (with the <img> element), videos (with the <video> element), sounds or music (with the <audio> element), etc.…

E-n displaying other HTML documents using the <iframe> element.

1.5.2. The uses of URLs are extensive

Other web technologies like CSS or JavaScript use URLs intensively.

1.5.3. How Do URLs Work?

When we enter a URL into an address box, your web browser employs a DNS (Domain Name Server) to convert the URL to the associated IP address. The browser may then utilize those numbers to navigate you to your desired location. [20]

1.5.4. URLs that are encrypted

A URL that begins with "https://" denotes that you are on a secure site. That means that any personal information you input on that site will be encrypted before it is transferred. Hackers cannot readily intercept encrypted information [21].

1.5.5. URL Manipulation Hacks in Web Application

An automated input hack modifies a URL and transmits it back to the server, instructing the web application to perform things like redirect to third-party sites, load sensitive files from the server, and so on. One such vulnerability is local file inclusion.

This is when a web application receives URL-based input and provides the contents of the given file to the user [22].

1.6. Conclusion

In this chapter, we talked about three main concepts which are the URL structure and the different types of malicious web content, as well as all the events of the attackers at the time of the COVID-19 virus.

These concepts are the basis of our work on malicious web content under the heading of COVID-19.

CHAPTER 02

RELATED WORK

2.1. Introduction

Although the problem of COVID-19 themed malicious websites has not been entirely resolved until now with the spread of the Corona epidemic, web content security has become the main goal for developers to aim to work remotely without attacks. In this chapter, we will provide the latest works on COVID-19 attacks to secure the content received by web browsers.

2.2 Approach 1

M.M. Ahsan Pritom, K.M.Schweitzer et al [12] proposed an approach to secure web content based on data-driven characterization, and detection of COVID-19 themed malicious websites, the experiments show that Random Forest classifier can detect COVID-19 themed malicious websites based on the lexical and WHOIS features.

The proposed methodology is decomposed into five modules: data collection, feature definition and extraction, data pre-processing, classifier training, and classifier test, as it is shown in figure 2.1.

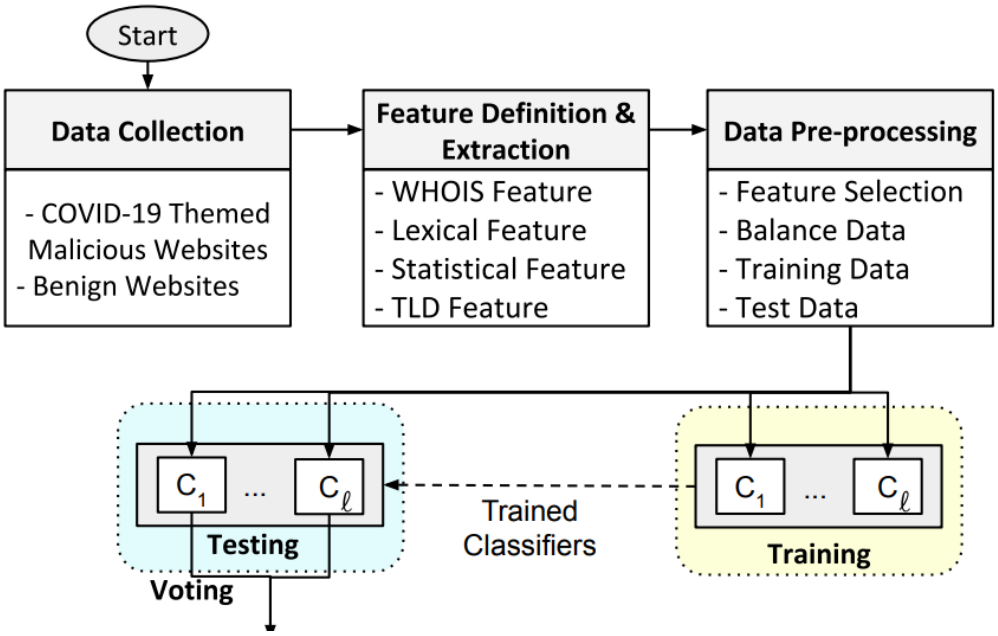


Figure 2.1: Approach general structure.

2.2.1 Data Collection

The collected dataset of COVID-19 malicious website are obtained from what was published between **2/1/2020** and **5/15/2020** by two sources:

- **(i)** Check-Phish [23], which contains 131,761 malicious websites waging scamming attacks related to COVID-19.
- **(ii)** Domain-Tools [24], which contain 157,579 malicious websites waging malware, phishing, and spamming attacks related to COVID-19.

The benign websites are obtained from top 250,000 websites from Cisco's Umbrella 1 million websites dataset on 16/05/2020, which is a source of reputable websites.

2.2.2 Feature Selector

This component is responsible for extracting the Features of malicious web content themed COVID-19.

The main step in this work is the selection of features, this model uses 11 features, which are new or modified and the others are obtained from previous works, they define features according to the following aspects of websites: WHOIS (F1- F4), domain name lexical information (F5-F9), statistical information (F10), and Top-Level Domain or TLD (F11).

2.2.3 Data Pre-Processing & Learning & Model Selector

It needs a data set to train the model and select the classification algorithm that gives the best results.

2.2.4 Detector

It uses the training model from the previous step to make the classification.

COVID-19 themed malicious website detectors must consider WHOIS features; and Random Forest performs the best among the classifiers that are considered.

2.3 Approach 2

J.Ispahany, R.Islam et all [14] proposed another approach to secure web content against phishing; they propose a framework for detecting malicious domain names with keywords associated with COVID. Using only 5 lexical features.

The detection of malicious links via the lexical features within the URL has been shown to be fast and is low risk since it does not require navigation into the malicious link.

2.3.1 Detection Model

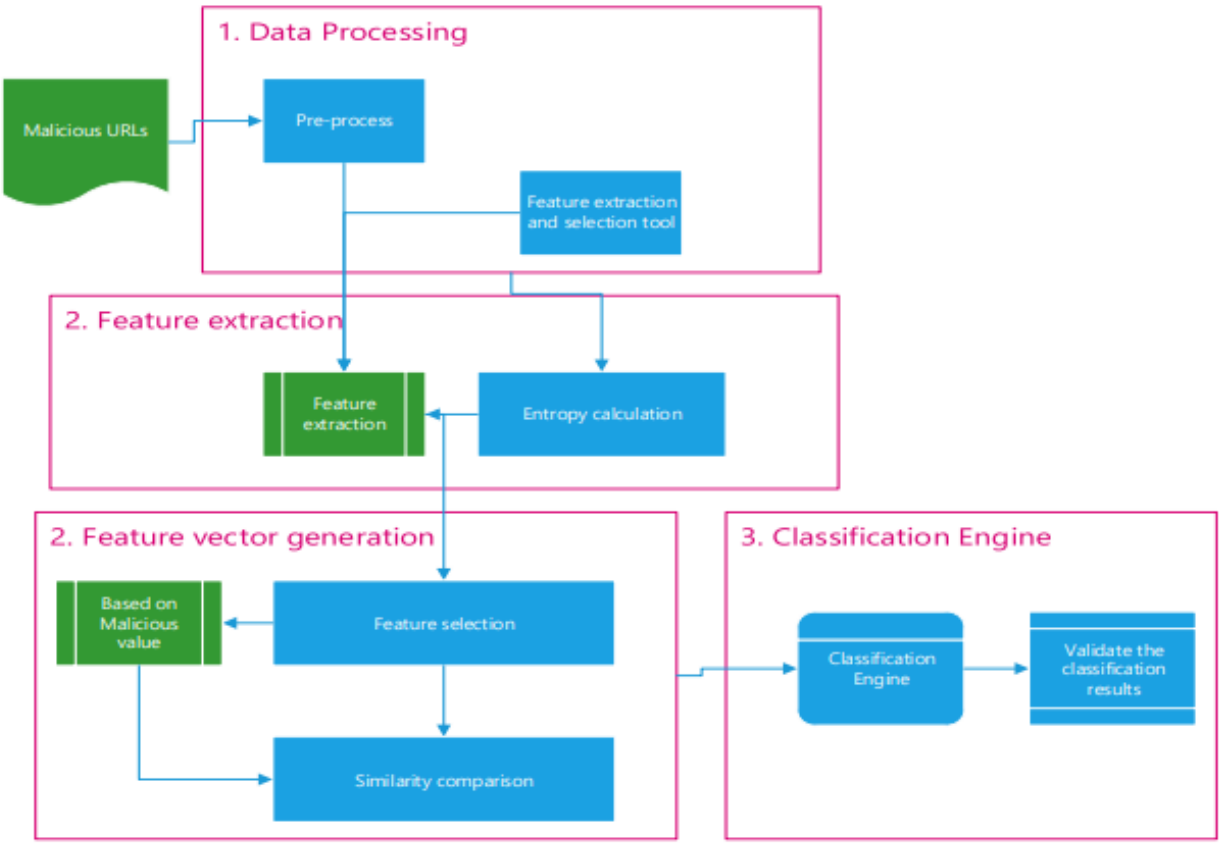


Figure 2.2: Proposed detection model [14]

2.3.2 Data pre-processing

2.3.1.1 Comparison engine

This study focuses on discovering URLs related to COVID-19. Therefore, the form searches for COVID-19 related URLs that are registered for malicious purposes. Initially, URLs (both malicious and benign) are absorbed into a comparison engine that searches for keywords related to COVID-19 and vague variants such as:
C0vid, Cov1d, and C0ronavirus.

2.3.1.2 Process the data set

After locating domain names related to COVID-19, all incoming URLs must be standardized to define the feature. The scope of this study focuses on detecting domain names registered for malicious intent.

To combat the influx of malicious URL's related to the coronavirus, they propose a mode which detects malicious URL's related to COVID-19. The detection of malicious URL's via the lexical features present in the hostname is fast and low risk since navigation into the domain name is required. Most detection models throughout literature have been designed to detect URL's from popular blacklisting sites such as Phish-Tank.

2.3.3 Features extraction

A. Lexical features

The lexical features have been used with success in all malicious URL detection studies. The only characters allowed at the time of domain name registration include 26 letters of the alphabet (A-Z), numbers (0-9), and hyphens excluding the beginning and end of the domain name string. Although previous studies included special characters such as percentage (%), curly braces ({}), or hash (#) as part of their features, they are not available at the time of recording. Therefore, no special characters were used throughout this study to detect malicious links.

URL Components removed during processing:

1) **Protocol** – Application layer protocol used to access the internet services. For example: http, https, ftp, etc.

2) **Ports** – The communication port used to access the service. For example: 80, 443, etc.

3) **Path** – The destination of the file on the target server. For example, path/file.php

4) **Query** – Request parameters forwarded to the target server. For example, userId=01

2.3.4 Feature selection

URL Components retained and used for feature selection:

a) **Domain name –** The registered identification string. For example, google, apple, amazon, etc.

b) **Top Level Domain (TLD) –** Domains at the highest level of the domain name system (DNS). For example, .com, .edu, .net etc.

c) **Second Level Domain (SLD)** – Domains directly below the TLD, for example .co, .au, .nz, etc.

d) **Length of domain name** – The total number of characters in the domain name are included as a feature. All other components such as TLD and SLD are not factored into the length calculation.

e) **Count of Hyphens** – The total number of hyphens present within the domain name. For example, www.examplewebsite.com contains one hyphen character.

f) **The total number of numeric characters** (from 0 – 9) present in the domain name. For example, **examp1e-webs1te.com** contains 2 numeric characters.

Table 2.1 : List of lexical Features used in this study

Number	Feature	Description	Previous studies using this feature
1	Length	Length of the domain name	[3][13][23][24][40]
2	-	Number of hyphens within the domain name	[13][24][40]
3	[0-9]	Number of numeric characters within the domain name	; [13][24][39]
4	Entropy	Shannon's entropy calculation of the domain name	[13][21]
5	Malicious value	Binary rating. 1 for malicious URL's and 0 for benign URL's.	N/A

2.3.5 Classification

For site classification the approach is based on three Machine Learning Algorithms which are: K-Nearest neighbor (KNN), Support Vector Machines (SVM) and Random Forest.

In this classification process, we give the generated feature vectors the WEKA classification system. In all experiments, 10-fold cross validation is applied to ensure a thorough mixing of features.

Using KNN algorithm without the entropy calculation in the dataset produced the highest accuracy (99.2%) and the lowest false positive rate (2%).

2.4. Approach 3

This approach is proposed by M.Houd, R.Djheiche et all [25] suggest method for developing a chrome extension that detects malicious web content. The extension relies on machine learning to identify malicious websites.

This approach takes two steps:

2.4.1 Features Extraction

They choose specific features extracted from URL and Web Page that help the extension to classify web sites, the web sites have a huge number of features for this reason they reduce the number of features in order to optimize the execution time of the extension.

The collection of features is divided into categories as bellow

2.4.2 URL Lexical Feature

Lexical feature is the textual properties of the URL itself. This feature list includes the length of the hostname, the length of the entire URL, the number of dots in the URL and token in the hostname (delimited by '.'), in the path URL (strings delimited by'/', '?', '.', '=', '-' and '_'). All these features are integer valued and their values range helps in identifying the malicious web URLs. For example: In Phishing attack the malicious URL has mimic appearance to the benign web page and only differs by these bag-of-words.

The table below presents the list of extracted features:

Table 2.2: List of extracted Features in this study

Features	Explanation
Double slash redirection	Detecting if there are double slash redirecting in the URL
Longest domain length	Calculate the Length of the Domain
Number of dots	Calculate the number of dots in the URL
URL length	Calculate the Length of the URL
host length	Calculate the length of the Host
URL Scripts	Detect the scripts and commands in the URL
sub-domain(www)	Check the type of the Sub-Domain
HTTPS	Check the HTTPS protocol
domain (.com,.net…)	Check type of the Domain
numbers in URL	Detect if there is numbers in the URL
count number	Count numbers in The URL
count letter	Count letters in the URL
rank host	Check the rank of the Host
Googlesearch engine	Check the URL by Google Search Engine
JavaScript in URL	Detect JAVA Script in the URL
malicious scripst in content	Detect the malicious scripts in the web page
external links	Detect if there are external links in the web page
iframe count	Count the I frames in webpage

2.4.3 Choice of Features

In this study, it selects a set of relevant features to identify harmful content. The choice is made according to the influence of the features on the results obtained.

The significance of the features is calculated using the average accuracy method which calculates the effect of each feature on the accuracy of the classifier

After the tests, 6 features that had no effect on the rating results were rejected.

2.4.4 Trained module generation

This work is based on analyzing both HTTP request and response, which make it able to detect the advanced malicious content cached on the web page if the URL analyzing result is benign. As described in the following flow chart (figure 2.2):

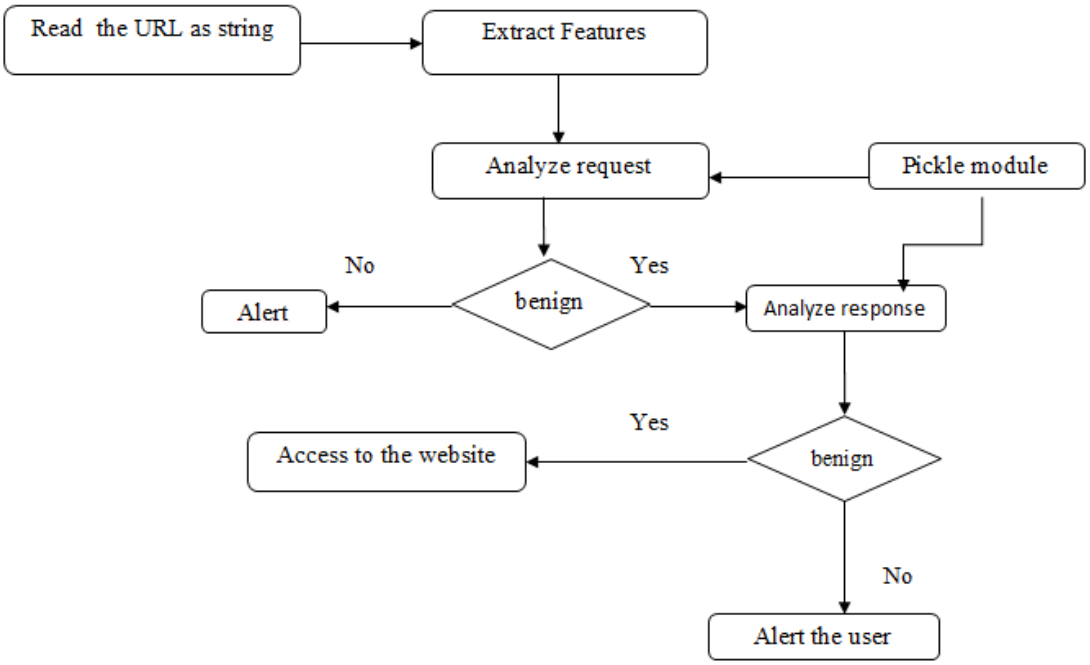


Figure 2.3: Web Guard flowchart [25]

2.4.5 Classifier

the extension is tested via four machine learning algorithms which are: Support Vector Machine (SVM),Naive Bayes(NB),K Nearest Neighbors(KNN),Random Forest Algorithm(RF).The selection based on the best results of precision sorted by the previous algorithms.

Unlike the previous works this classification approach is based on a module generated in training phase as .PKL file.

This module stores the data of the training phase so in new classification operation the classifier predicts directly from the module doesn’t need a new training session which is a positive point to reduce the process time.

2.5 Comparison of previous works

In this section we would make a comparison of the works that we reviewed in this chapter. Our comparison based on the common characteristics of those works.

Table 2.3: Approaches Comparison

	Approach 1	**Approach2**	**Approach 3**
URL analysis		×	×
Web page analysis	×		×
Type	study on characterizing and detecting malicious websites	Detecting Malicious URLs Using ML	Google Chrome extension
Detected content type	COVID-19 themed malicious websites	COVID-19 Malicious website	Malicious website content in general

In this comparison we notice that the approach3 is based on URL and page content features, it is a Google Chrome extension which can be a generic model to our work Approach 2 is a google chrome extension against phishing based on Lexical URL features , those features are important in our work

Approach 1 is the work uses webpage content features and URL analysis in the classification which presents a study on the characterization and detection of malicious websites on the topic of COVID-19.

2.6 Conclusion

In this chapter we talked about the previous works, explained their characteristics and functionalities and finally make a comparison between them to propose our own extension in the next chapter.

CHAPTER 03

CovProtectWeb Extension Against COVID-19 Malicious Content

3.1 Introduction

After reviewing the previous works for identifying harmful web content related to Corona virus, we suggest our strategy for developing a chrome extension that identifies dangerous online content in this chapter. Our extension uses machine learning to detect fraudulent websites, regardless of whether they have COVID-19 as a theme or are impacted by COVID-19.

3.2 Our work

Most Internet users are subjected to dangerous online content when using the web, particularly during the Corona virus season. As we saw in the last chapter, the techniques utilized disclose a wide range of potentially hazardous features.

So, to make browsing more safe on the client side, we propose the CovProtectWeb extension for Google Chrome. It examines URLs queries and responses from websites to detect harmful online content labeled COVID-19 or infected with the virus.

3.3 Idea of CovProtectWeb

CovProtectWeb is a Google Chrome extension that intercepts all Chrome browser traffic. CovProtectWeb scans web site requests in order to detect dangerous information on websites.

CovProtectWeb is made up of three primary components: the feature extractor, training module, and classifier.

Fig 3.1 shows our extension architecture which contains three main modules:

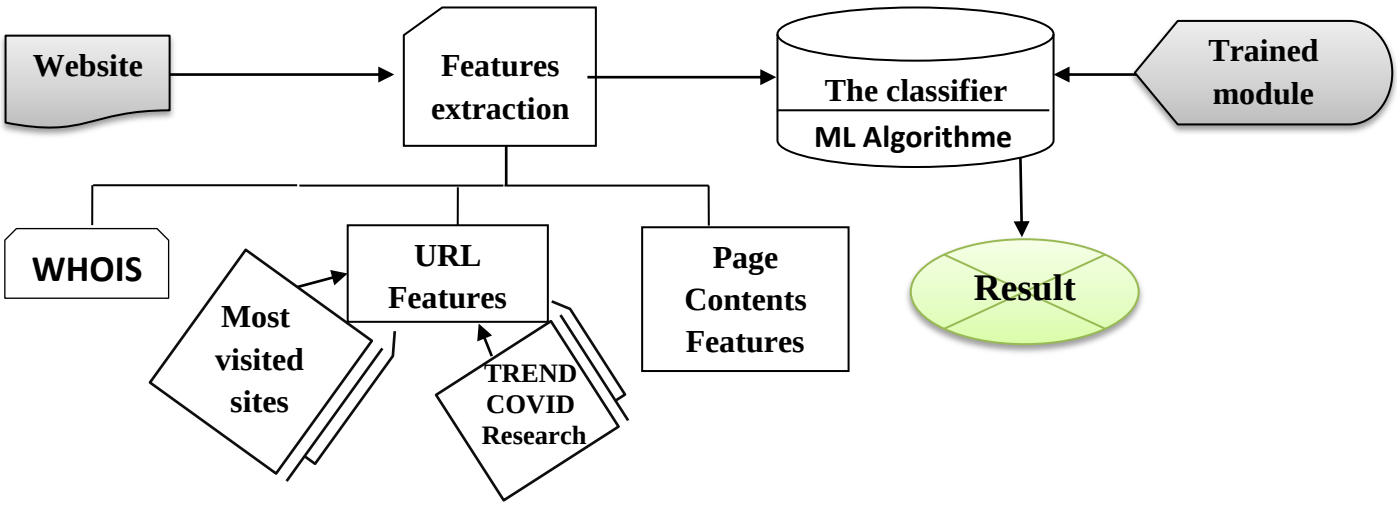


Figure 3.1: Architecture of CovProtectWeb extension in general

3.3.1 Feature Definition and Extraction

We choose specific features extracted from a URL and a webpage that helps our classifier to categorize the website. Websites have a large number of features; we propose to categorize them into four classes:

- WHOIS features.
- Web browsing features
- URL lexical features
- Web content features

We choose the relevant features from previous works and our own studies:

In our work we combine some of the features that have been extracted in previous studies in just one feature to get better results.

For example, but not limited to, we combine the length of the URL and the length of the host name so that the length of the URL is not more than 25 characters, while the length of the host that is inside the full URL is not more than 12 characters.

(Url-length_And_Host-length)

After studying all the previous features we can analyze the URL by combining features are always in the headlines harmful.

3.3.1.1 WHOIS Based Feature

It represents a set of useful information that can be found in WHOIS database. We can find out when the domain expires, what is the transfer status! , who is

registered for the domain! , who should be contacted in case of abuse! , and what name servers are used!
Where in the period of the spread of the corona virus was registered new domains bearing the theme of covid-19 because it is a new event and are harmful domains as mentioned

We also note that most of the malicious sites that were registered in the period of covid-19 are dying fast, so we must note the date of Registration, date of update, date of end of registration and from there we judge the site harmful or otherwise.

The selected WHOIS features are:

• Current WHOIS Registration Duration and Remaining Expiry Period: We mean the number of days since the website was registered, and the number of days left before a website's registration expires, to the date on which the new event occurred and extract the value of this feature by date. in relation to the short date on which the website is registered and compare the value of this feature

• Number of days since the website was last updated and Remaining Expiry Period: Here we mean the number of days since the website was last updated with and the number of days left before a website's registration expires respect to the date a new event appeared and extract the value of this feature to the short date on which the website is registered (In the case of our study, we always compare the date of the emergence of the Covid-19 virus)

In our work we combine some of the features of WHOIS.

3.3.1.2 Browsing Features

Attackers take advantage of Internet users' orientation to obtain information like google trend and ALEXA ranking
However, there is still oscillation. One possible cause is that the attackers have been waiting to create new COVID-19 themed malicious websites based on the pandemic's new developments (e.g., vaccine).

The most extracted features for browsing are:

A. **Ratio and Distance of similarity between the URL with the ranking site DZ:** We extract the most visited sites anywhere (our example in Algeria) from ALEXA where we can buy 500 sites and in our study we extract 50 sites by Web

Scraping, then we compare the similarity between each site and the URL under test ,because this feature identifies phishing sites is done by the Levenshtein distance algorithm,

As a hacker circumvents, we circumvent it.

B. **Check the rank of the Host in ALEXA:** We also extract the site ranking if this URL after the first million sites it is a site where there is little doubt that it is harmful

C. **Check the URL by Google search engine:** We extract the SEO site ranking if it is in the first three pages it means that the site is benign and enters a lot of users

D. Check the TREND research related COVID: Similarity between URL and Research Trend related mot COVID extracted with associated queries by Google TRENDS.

3.3.1.3 URL Lexical Feature

The lexical features of the URL themselves are the lexical characteristic . Our extracted features are:

a. Lexical features in URL

E. The length of the host name, as well as the total length of the URL,

F. The number of dots contained in the URL Token (delimited with ".") The domain name "**univ-msila.dz**" contains one (1) dot [.] as a character.

G. The number of domain hyphens in the path's URL (strings denoted by '/', '?', '.', '=', '-').

H. The number of vowels equals the number of letters in the title words because they will be nested in the domain name (e.g., a, e, I o, u).

I. Numbers Ratio in the domain: This is the proportion of the number of numbers (0-9) in a field name to the total amount of characters including numbers.

J. The number of distinct alphanumeric characters in a field is: This is the total amount of distinct alphanumeric characters in the domain name (for example, a-z, A-Z, 0-9).

b. Lexical features in phishing URL

Addresses that are similar to the most visited site: In a phishing attack, for example, the malicious URL looks identical to the benign web page and differs only by that collection of words such as:

1. http:///www.net-flix.com
2. http://signin.netflix.com@12.3/
3. www.gofornetflix.com

The above phishing URL appears to be identical to the benign URL https://www.netflix.com/home.

The string "net", "flix", and "com" appear in all three URLs above, making the malicious URL appear similar to a benign URL and allowing the user to easily slip into the attacker's network.

All of these features are assessed using an integer, and the range of their values aids in the identification of URLs class.

3.3.1.4 Features in Page Content

The source code of a web page is a valuable source of information for spotting fraudulent web pages. Web pages' contents-based features include a variety of functions, objects, Frames, data streams, hyperlinks, In-bound, Out-bound, and hidden components.

These features are based on the weaknesses and flaws in the scripting language. HTML, CSS, and JavaScript are examples of scripting languages.

a. JavaScript features:

JavaScript is a common scripting language that is utilized as the web page validation code. Some JavaScript functions are extremely vulnerable and are used by the attacker to inject malicious material(or XSS). By just clicking on the page, these malicious scripts are run. Some of the most commonly used JavaScript functions are as follows: eval(), escape(), unescape(),exec(), ubound() etc.

3.3.1.5 Safe Browsing

Safe Browsing is a Google tool that checks URLs against Google's regularly updated lists of hazardous web resources. Social engineering sites (phishing and deceiving sites) and sites that include malware or unwanted software are examples of dangerous web resources. Come take a look at what's possible.[26]

We use Safe Browsing, which allows us to do the following:

• Check pages against Safe Browsing lists based on platform and threat kind.

• Inform users before they click on links on your website that may lead to compromised pages.

• Prevent users from providing links to your site's known contaminated pages.

The table 3.1 presents the list of extracted features:

Table 3.1: List of features

Features Definition	Features Explanation
WHOIS Based Feature	* Current WHOIS registration lifetime * Remaining WHOIS expiration lifetime * Number of days since last WHOIS update
Features Browsing	* Ratio and Distance of similarity the URL with the ranking site DZ * Check the rank of the Host in ALEXA * Check the URL by Google search engine * Check the TREND research related covid19
Lexical features in URL	* Detecting if there are double slash redirecting in the URL * Calculate the Length of the Domain * Calculate the number of dots in the URL * Calculate the Length of the URL * Calculate the length of the domain and length of the URL * Scripts and commands in the URL * Check the HTTPS protocol * Count numbers in The URL * Count letters in the URL * Detect JavaScript in URL * Count numbers and count letters in The URL * Similarity between URL and names Research Trend related COVID-19

Features in Page Content	* Detect malicious scripts in the web page * there external links in the web page * Count the I-Frames in webpage * Detect Research Trend related COVID-19 in the web page

3.3.2 Features Selection

To choose features related to identifying harmful content on the theme of the COVID-19 virus in our analysis. We provide the outcomes of machine learning processing of prior data and obtain the findings based on the degree of influence of the features.

The validity of features is calculated using the average accuracy approach, which computes the effect of each feature on the classifier's accuracy.

The main accuracy approach is a direct measurement of the impact of each attribute on the model's accuracy. The fundamental concept is to change the values of each feature and see how much permutation reduces the model's accuracy.

Obviously, switching should have little or no influence on model accuracy for non-significant variables, whereas switching significant variables should significantly reduce it.

Following the tests a lot of features that had no bearing on the rating results. That is, the average accuracy is close to 0 degrees.

Among these features we mention:

- Full URL character count feature (accuracy rate 0, also useless with benign URLs with more characters) - Alphabet count feature (0 accuracy rate)
- Also, the advantage of having external links on a web page is not a criterion for accuracy.
- Check the URL by Google search engine.

3.3.3 Data Pre-Processing

Since websites do not contain information about the features we have chosen, it is important to consider different scenarios. In our example, we suggest looking at two data sets that can be derived from Google because some websites do not contain information about WHOIS features.

First, in the analysis stage of the previous data, which are the benign and malicious sites, we appropriately analyze the URL to extract a list of relevant features , and convert it into a numeric vector , Then, after converting the URL to a numeric vector, we create a new file to store the URL as a digital vector by extracting the features from the URL by calling the extraction function (Feature Selection Function), which returns to us each feature in the form of a numeric value (either 1:Bengnin Value, -1:Malicious Value, or 0:Suspecious Value).

This final vector is saved in the created file. As a result, this stage comprises of four major phases:

- we appropriately analyze the URL by convert it to string
- extract a list of relevant features ,
- we convert it into a numeric vector
- Then, we create a new file to store the URLs as a digital vectors

3.3.4 Dataset construction

We created a dataset of 400 sites in order to train the workbook. Our dataset contains 100 secure Alexa top Sites and 100 secure covid-19 related sites, it also contains 100 dangerous sites that carry the harmful word COVID-19, And 100 sites are generally malicious.

Since the Dataset only contains some WHOIS information and some other feature information, we use it to study feature selection. For this purpose, we use the forest random classification feature importance method (With our breakdown of training test data) to find the important features. The relative importance of the features is placed. We note that some features have very small relative significance (i.e. < 0.01) when compared to others, indicating that (e.g. the hyphens feature and the numbers feature are used equally in malicious or benign domain names). Thus, we will remove them in the study.

Figure 3.2: Dataset: dangerous sites that carry the harmful word COVID-19

3.3.5 Training phase

For good classification results among four machine learning algorithms namely: support vector machine (SVM), Naive Bayes (NB), nearest neighbors (KNN), random forest algorithm (RF). Characterization study and detection study conducted show that Random Forest classifier can detect COVID-19 themed malicious websites based on lexical features and WHOIS.

We choose in our extension the random forest algorithm based on the best results of accuracy as per previous algorithms in previous works

3.3.5.1 Cross validation

A Cross-Validation is an essential step in the Machine Learning process. We can't just assume that our model will operate well on data it hasn't seen before once we've finished training it. To put it another way, we can't guarantee that the model will have the appropriate accuracy and variance in a real-world setting. We need some type of proof that the forecasts our model generates are accurate. We'll need to validate our model for this Validation refers to the process of determining if numerical results measuring hypothesized correlations between variables are acceptable as data descriptions. [27]

a. Split method for Training set / Testing set

In this method, the entire data set is randomly divided into training and test sets. After that, train the model on the training set and validate it on the test set, ideally splitting the data 80% - 20%. If we just have minimal data, this method may result in a high level of bias.

b. The K-Folds method

The K-Folds technique is simple to understand, and especially popular. Compared to other Cross-Validation approaches, it usually results on a less biased model.

For good reason, it ensures that all observations from the original dataset have the chance to appear in the training set and in the test set. In case of limited input data, this is therefore one of the best approaches.

First, we start by separating the data set randomly into K folds. The procedure has a unique parameter called ' K ' referring to the number of groups into which the sample will be divided.

The value of K must not be too low or too high, and a value between 5 and 10 is generally chosen depending on the size of the dataset. For example, if K=10, the dataset will be divided into 10 parts.

A higher K-value leads to a less biased model, but too wide a variance can lead to over-adjustment. A lower value is equivalent to using the Split Train-Test method.

The model is then adjusted using the folds K-1 (K minus 1). The model is validated using the remaining K-fold. Scores and errors should be noted.

The process is repeated until each K-fold serves within the drive assembly. The average of the scores recorded is the performance metric of the model.

In other cases, the K-Fold technique is used by default to split and train the model. Folds can be used as iterators or in a loop to perform training on a Pandas data frame.

3.3.5.2 Pickle module

Unlike the previous works our classification approach is based on a module generated in training phase as .PKL file.

The PKL file type is primarily associated with Pickle by Python. Pickle is a module of Python that serializes objects so they can be saved to a file, and loaded again when called by the program. Serialization of files is called pickling and

deserialization is called unpickling. Files that are serialized by the Pickle module are stored in the PKL format.

This module stores the data of the training phase so in new classification operation the classifier predicts directly from the module doesn't need a new training session which is a positive point to reduce the process time.

3.3.6 Testing phase

In our work, the extension first analyzes the URL and page content, after the extraction of:

- The most visited sites in Algeria as an example of our study.
- The trend Google Covid name

After reading the URL as a string we compare its similarity with the most visited sites and trending words.

As we read the URL content, we extract all its features

Both URL and content are analyzed to be tested by pickle model

If the site is secure, the user is notified to continue, but if otherwise, the user is alerted to not continue. As illustrated in the diagram below (Fig. 3.4):

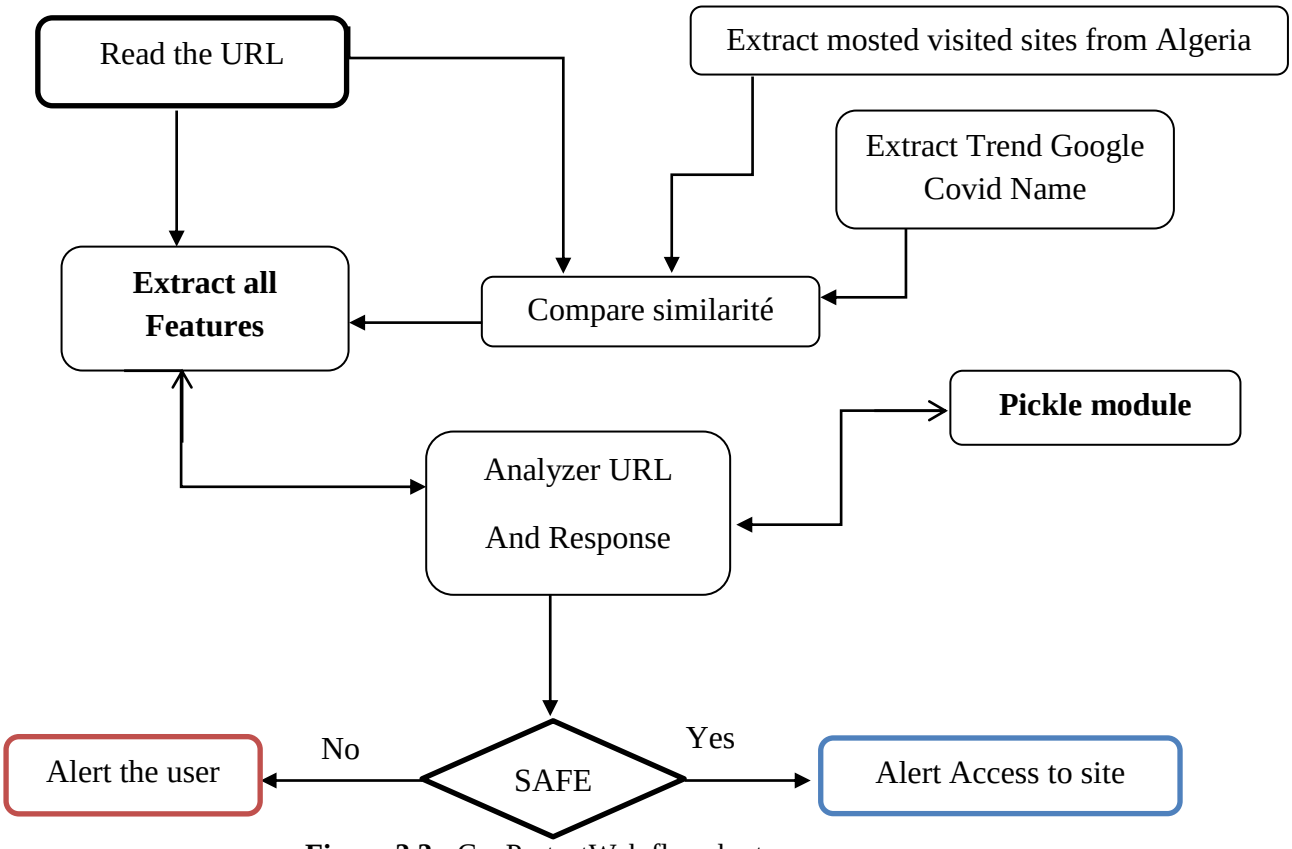


Figure 3.3: CovProtectWeb flowchart

3.3.6.1 Random Forest is one of the four algorithms we use

Here are some reasons why we employ the Random Forest algorithm:

It requires less training time than other algorithms, predicts output with high accuracy even for big data sets that run efficiently, and can maintain accuracy even when a considerable percentage of data is lost.

3.3.6.2 Random Forest Algorithm

Random Forest [28] is a well-known machine learning algorithm from the supervised learning technique. It can be applied to both classification and regression issues in machine learning. It is built on the notion of ensemble learning, which is the process of merging numerous classifiers to solve a complex problem and improve the model's performance.

"Random Forest is a classifier that contains a number of decision trees on various subsets of the provided dataset and takes the average to enhance the predicted accuracy of that dataset," as the name implies. Instead than relying on a single decision tree, the random forest collects the forecasts from each tree and predicts the final output based on the majority vote of predictions. the greater the number of trees in the forest, the higher the accuracy and the lower the risk of overfitting.
The Random Forest method is depicted in the diagram below **:**

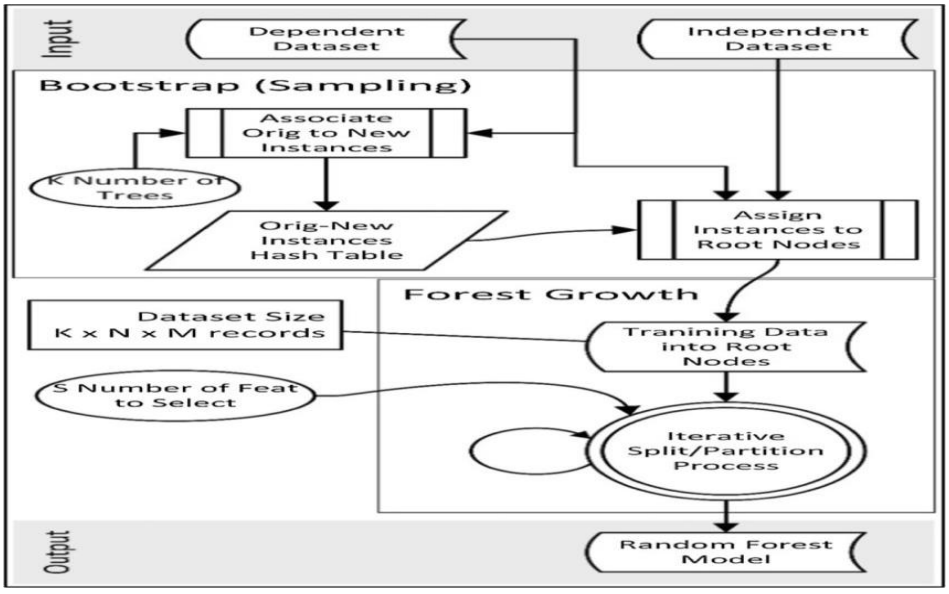


Figure 3.4: Random Forest initial implementation

3.4 Conclusion

In this chapter we present the implementation of our web extension named CovProtectWeb and describe its components, as well as the used techniques to protect web users from hazardous content related to the COVID-19 topic of website.

In the following chapter, we will show the procedures for implementing CovProtectWeb, as well as the tools used in implementation, and we will discuss the outcomes of trials.

CHAPTER 04

Implementation & Experimentation

4.1 Introduction

In this chapter, we present the steps of implementing and testing our web extension that we explained in chapter three, which is developing a web browser extension for Google Chrome to detect malicious web content related to COVID-19. We explain the implementation tools and experiments to test and measure the performance of our extension.

4.2 Development tools

We use to develop our web extensions the following tools:

4.2.1. Google chrome browser

We choose Google Chrome browser to conduct our study because Chrome is the most used browser according to global statistics in the most famous outbreak of COVID-19 64.73% (in Figure 4.1), and also because it is an open source browser..

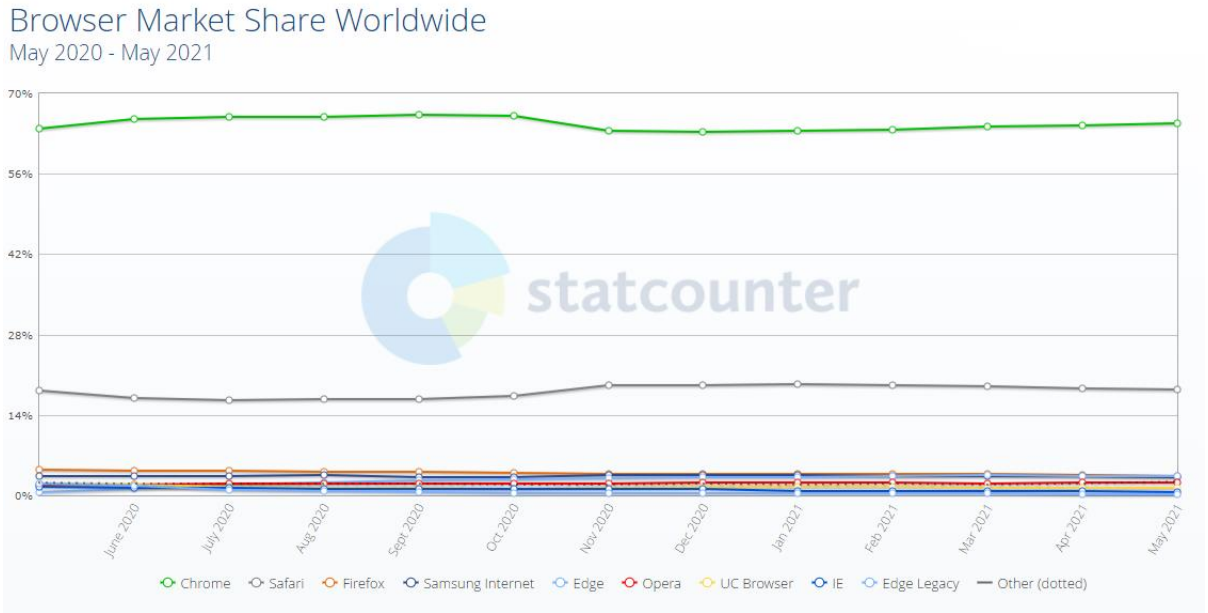


Figure 4.1: Stat counter Global sites Browser Market Share Worldwide [30]

4.2.2. Chrome Developer Tools

Chrome Developer Tools is a comprehensive toolkit for developers, built directly into the Chrome browser. These tools let we edit web pages in real time, diagnose problems more quickly, and build better our extension faster.

Though, we can still get some use out of Developer Tools.

We were using Developer Tools for:

- Identify load speed issues.
- Catch non-working plugin. There easy way to catch it in Developer Tools.

We Open the site were want to check; we open Developer Tools and go to the Console tab.

- Saw what technology a site is using

We selling tech, but we don't know if this work is a potential customer. Some technographic information would be nice. Let's we open Developer Tools and get some. With Developer Tools open in the Elements tab, find the <head> tag in the HTML and click on it to expand the section. [31]

4.2.3. PHP Language

PHP is an HTML-based server-side scripting language. We use it to track sessions and manage dynamic content.

PHP that is tolerant: PHP makes every effort to be as forgiving as possible.

PHP is fast to execute, especially when compiled as an Apache module on the Unix side; once begun, it executes with large result sets when indices are being created.

We can utilize PHP to prevent users from accessing your website page.

PHP works with practically every web server software, however we'll utilize whichever Apache server is available for free. [32]

We used PHP interpreter as an intermediary between JavaScript in the Extension and the code python that using ML techniques.

ClientServer.php

```php
<?php
//header('Content-Type: application/json');

$url = $_GET['url'];

echo $url;

file_put_contents('markup.json', json_encode(array('url' => $url, JSON_PRETTY_PRINT)));

$decision=exec("python teting.py");

echo $decision;
```

Figure 4.2: Code PHP for interpreter URL

As a web server, we use Apache.

4.2.4. Machine Learning in Python

Predictive modeling is a sub-discipline of machine learning. This is the most useful sector of machine learning in industry, and it's the form of machine learning that Python's scikit-learn package excels at assisting. Unlike statistics, where models are used to make sense of data, in data science, models are used to make sense of data. Predictive modeling focuses solely on creating models that generate the most accurate forecasts at the expense of explaining why such predictions are made. Unlike the broader topic of machine learning, which may be applied to any type of data, predictive modeling is solely concerned with tabular data (e.g. tables of numbers like in a spreadsheet).[33]

We get started and using Python for applied machine learning effectively and quickly.

I. Why Python?

- Python works on different platforms (Windows, Mac, Linux Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

II. Data Requirement Pre-processing in python

Almost all of our data must be pre-processed. It is a necessary step. One issue is that different algorithms make different assumptions about our data, necessitating the use of alternative transforms. Furthermore, when all of the requirements are followed and our data is prepared, algorithms can occasionally produce better results without pre-processing.

In general, we propose establishing a number of various views and transforms of our data, and then running a number of algorithms on each view. This will aid us in determining which data transforms are best for revealing the structure of our problem in general. [34]

We also convert the good URLs and add them to the category manually with an increment of 1, And we convert the malicious urls and add the category to them with an increment of -1

This is how we get the set of data that we give to machine learning so that it can train on it. As indicated in the figure: 4.3

generait.py

```
import f_url
from csv import writer
from f_url import LOCALHOST_PATH, main

def append_list_as_row(file_name, list_of_elem):
    with open(file_name, "a+", newline='') as write_obj:
        csv_writer = writer(write_obj)
        csv_writer.writerow(list_of_elem)

fields = ['https1', 'double_slash_redirecting2', 'largest_domain_length3', 'number_of_dots4', 'url_length5', 'host_leng
          'rank_host12', 'javascriptinahref13', 'xss14', 'externallinks15', 'iframecount16', 'similarityUrlTopSite17',
# ListTestBadeSite after finally run these
# site_bade.txt
file = open("site_safe.txt", "r")
for url in file:
    features_test = f_url.main(url)
    append_list_as_row("ListTestSafeSite.csv", features_test)
```

Figure 4.3: Code Python for generate Dataset

III. Web Scraping

Web scraping is the process of extracting specific information as structured data from HTML/XML content. We often need to fetch and extract data from many websites to build data sets and test or train algorithms, neural networks, and machine learning models.

The website usually offers APIs which are a great way to fetch structured data. However, there are times when the API is not available or you want to bypass the registration process.

Under our working conditions, the data can only be accessed via the web page (data about the statistics of the most visited sites). The manual process can be quite cumbersome and time consuming when dealing with dynamic website related data that must be accessed frequently. Python offers an automated way, through various modules, to fetch HTML content from the web (URL/URI) and extract data. We will do a web scraping using the beautiful soup module. [35]

Information extraction is used for search engines, news libraries, manuals, domain-specific text or dictionaries. A form of information extraction is text mining, an information retrieval task aimed at discovering new, previously unknown information, by automatically extracting it from different text resources [36]

A more recent variant of Web crawlers are Web scrapers, which are aimed at looking for certain kinds of information–such as prices of particular goods from various online stores—extracting, and aggregating it into new Web pages

We focus on web scrapers that extract textual information from the web page that gives us the most visited sites in Algeria and in the latest modification. There are many ways to pull information from the web Although this is sometimes the only way to export information from a web page, this is not practically possible, as regular expressions are used to find information that matches some pattern. More web scraping techniques are HTTP programming, DOM parsing, and HTML parsers. Finally, the method of web scraping consists of going to sites to scrape them and then automatically creating a file in which the data is saved. It is worth noting that Web scraping may be against the terms of use of some websites. Being interested in the scientific issues concerned with the adoption of Web scraping to perform Web advertising, we do not take into account legal issues on adopting and implementing Web scraping techniques.

```
titles = []
url = 'https://www.alexa.com/topsites/countries/DZ'
response = requests.get(url)
src = response.content
soup= BeautifulSoup(src , "lxml")
#title = soup.find_all("div",{"class":"tr site-listing"})
x= soup.find_all('div',class_='td DescriptionCell')
for xxx in x :
    hand = xxx.find('a').text
    titles.append(hand)

with open(r"C:\Users\mokak\OneDrive\Desktop\EXTEnes test\testinger.csv","w", newline = '') as myfile:
    wr = csv.writer(myfile)
    wr.writerow(["titles"])
    wr.writerows([titles])

f1=open(r"C:\Users\mokak\OneDrive\Desktop\EXTEnes test\testinger.csv","r+")
input=f1.read()
input=input.replace(',','\n')
f2=open(r"C:\Users\mokak\OneDrive\Desktop\EXTEnes test\testinger.csv","w+")
f2.write(input)
f1.close()
f2.close()
```

Figure 4.4: Web scraping python Code

4.2.5. What is a Good Alexa Ranking?

Is a measure of the popularity of a website? It rates millions of websites in order of popularity, with an Alexa Rank of 1 being the most popular.

Alexa ranks websites from #1 to seemingly infinity. In 2020, there are about 1.5 billion websites that can be ranked by the Alexa algorithm. [37]

Google is the #1 website in the world according to Alexa rankings. The top 500 sites on Alexa are listed here for those interested in trying to crack the list.

While there is no exact range for what defines a good Alexa rank, generally, the lower the number, the better. It seems that any Alexa ranking of over 1000,000 is considered very high.

4.2.6. Good sites work on Alexa Rank and SERP

Alexa Rank also appears to have no effect on Google search.

From an SEO point of view, knowing where to rank traffic and where to post is important. Ranking priority is to focus benign sites on consistently providing the audience with amazing value and keeping up with best SEO practices.

In contrast to malicious sites, they are only up to date with new news, and therefore their appearance in the classification is with the latter. Also, malicious sites use a domain name that resembles the domain name of the top sites for search

4.2.7. Algorithm of Levenshtein Distance

The Levenshtein distance is a string metric used to compare two sequences. The Levenshtein distance between two words is the number of single-character modifications (insertions, deletions, or substitutions) required to turn one word into the other.

Edit distance is another name for Levenshtein distance, which is part of a larger family of distance measures. It is similar to pairwise string alignments.

The Levenshtein distance between two strings a, b (of length |a| and |b|, respectively) is given mathematically by leva,b(|a|,|b|), [38] where:

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 4.5: Mathematically, the Levenshtein distance between two strings a, b

where 1(a(i)b(i)) is the indicator function equal to 0 when a(i)b(i) and 1 otherwise, and leva, b(i,j) is the distance between the first I and first j characters of a and b.

It is worth noting that the first element in the minimum relates to deletion (from a to b), the second to insertion, and the third to match or mismatch, depending on whether the relevant symbols are identical.

4.2.8 WHOIS

We can use the WHOIS service to see who the registered owner of the domain name is. There is a Python module, called python-whois, for this protocol, which can be installed via pip using the pip install python-whois command [39]. It Works with Python2&3.

With the python-whois we can: [40]

- Create a simple importable Python module which will produce parsed WHOIS data for a given domain.
- Able to extract data for all the popular TLDs (com, org, net, …)
- Query a WHOIS server directly instead of going through an intermediate web service like many others do.

```
def is_registered(domain_name):
        w = whois.whois(domain_name)
        return bool(w)

##whois creation new
def expiredate(url):
    try:
        if is_registered(url):
            whois_info = whois.whois(url)
            date = whois_info.expiration_date
            ##  date expiration is very  So close
            if type(date) == list:
                date_expire = date[0] < datetime.datetime(2022, 1, 1, 0, 0)
```

Figure 4.6: example for Uses WHOIS library

4.2.9 Weka

Weka [41] is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

4.3 Experimentation setup

We run all our tests on a custom built computer while disabling all sort of processes that has may affect the performance one way or another.

- Operating System: Microsoft Windows 10 Pro 64 bit
- CPU: Intel i5-5210U @2.7Ghz
- RAM: 8 GB @3000Hz
- IDE: visual studio code x64-1.57.1
- Built in monitoring tool: Google dev tool
- Data mining tool: Weka 3.8.5
- Web browser: Google Chrome
- Server environment: Apache PHP 5.6.18
- Programming language: Python 3 ,PHP , JavaScript

4.4. CovProtectWeb Interface

CovProtectWeb is our extension which works on the browser's background with a simple interface and with click at the icon and wait (figure4.4)

Figure 4.7: CovProtectWeb Extension Icon

The analysis result is returned as it is illustrated in the figures bellow:

Figure 4.8: CovProtectWeb Extension example website safe

4.5 Experimentation

In this section, we will evaluate the performance of our extension by running it with our produced dataset.

We used the method of Cross-validation, it is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

According to previous study [12] [14]: the Random Forest classifier is the best ML model to detect COVID-19 themed malicious websites based on lexical and WHOIS features.

Therefore, below are two assumptions for a better random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

In our case, Figures bellow shows the obtained result by Weka ML tool.

- Figure 4.9 shows the obtained accuracy of 99.3% with an error rate of 0.7% using Random Forest.
- While Fig 4.10 and 4.11 shows the obtained accuracy: 98.8% , 98.5% using respectively SVM and KNN algorithms.

In our extension, we choose the random forest algorithm based on its best accuracy results.

```
Time taken to build model: 0.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         397               99.5    %
Incorrectly Classified Instances         3                0.5    %
Kappa statistic                          0.99
Mean absolute error                      0.015
Root mean squared error                  0.072
Relative absolute error                  3.0043 %
Root relative squared error             14.391  %
Total Number of Instances              400

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,997    0,007    0,993      0,997    0,995      0,990    1,000     1,000     1
                 0,993    0,003    0,997      0,993    0,995      0,990    1,000     1,000     -1
Weighted Avg.    0,995    0,005    0,995      0,995    0,995      0,990    1,000     1,000

=== Confusion Matrix ===

   a   b   <-- classified as
 199   1 |   a = 1
   2 198 |   b = -1
```

Figure 4.9: Random Forest Result in Weka

```
Time taken to build model: 0.20 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         393               98.8333 %
Incorrectly Classified Instances         7                1.1667 %
Kappa statistic                          0.9767
Mean absolute error                      0.0117
Root mean squared error                  0.108
Relative absolute error                  2.3333 %
Root relative squared error             21.6025 %
Total Number of Instances              400

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,993    0,017    0,983      0,993    0,988      0,977    0,988     0,980     1
                 0,983    0,007    0,993      0,983    0,988      0,977    0,988     0,985     -1
Weighted Avg.    0,988    0,012    0,988      0,988    0,988      0,977    0,988     0,983

=== Confusion Matrix ===

   a   b   <-- classified as
 198  2 |   a = 1
   5 195 |   b = -1
```

Figure 4.10: SVM Result in Weka

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         391               98.5    %
Incorrectly Classified Instances         9                1.5    %
Kappa statistic                          0.97
Mean absolute error                      0.0142
Root mean squared error                  0.1127
Relative absolute error                  2.8319 %
Root relative squared error             22.5334 %
Total Number of Instances              400

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,030    0,971      1,000    0,985      0,970    0,995     0,983     1
                 0,970    0,000    1,000      0,970    0,985      0,970    0,995     0,996     -1
Weighted Avg.    0,985    0,015    0,985      0,985    0,985      0,970    0,995     0,990

=== Confusion Matrix ===

   a   b   <-- classified as
 200   0 |   a = 1
   9 191 |   b = -1
```

Figure 4.11: KNN Result in Weka

4.5.1 Classification evaluation metrics

- The performance of our classifier may be measured using three key metrics or criteria: Calculation of Precision, Recall and Accuracy in the confusion matrix.

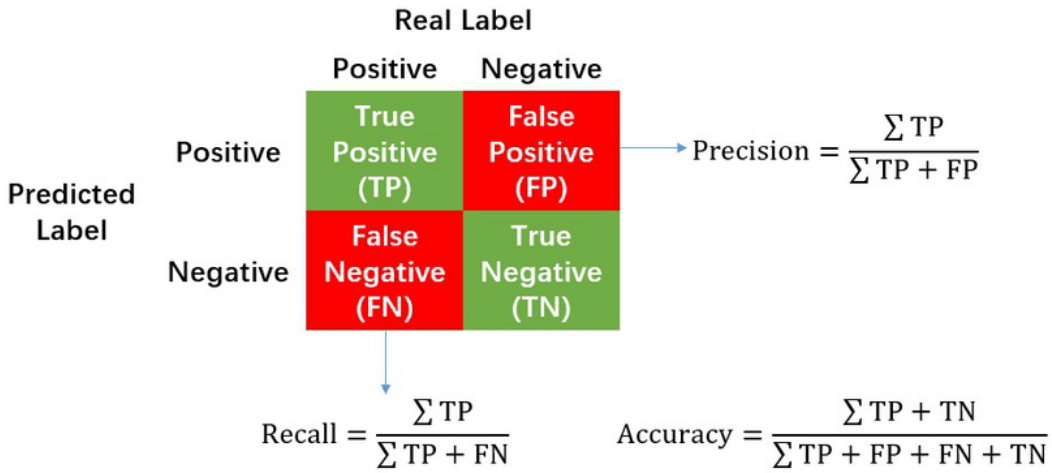


Figure 4.12: The performance of classifier formula [42].

Precision

The precision of a model in predicting positive labels is measured. Precision solves the question of how often a model was correct out of the number of times it predicted a positive outcome. The percentage of your findings that are relevant is known as precision.

$$precicion = \frac{TruePositive}{TruePositive + FaLsePositive}$$

Equation 4.1: Precision Formula.

Accuracy

Accuracy is a metric for evaluating a model that allows you to count how often of its predictions are correct.

Accuracy will provide an answer to the question of how accurate the model's predictions were. True Positives and True Negatives are examined by Accuracy.

$$Accuracy = \frac{TruePositive + TrueNegatives}{TruePositive + FaLsePositive + TrueNegative + FalseNegative}$$

Equation 4.1: Accuracy Formula.

- **Recall**

The percentage of actual positives that a model properly detected is calculated by recall (True Positive). When the penalty of a false negative is considerable, recall should be used.

$$Recall = \frac{TruePositive}{TruePositive + FaLseNegative}$$

Equation 4.2: Recall formula

The scores obtained for our extension's accuracy, recall, and precision where using three algorithms of classification which are the Random Forest, SVM, KNN, are presented in Table 4.1

Table 4.1: Results of the classification metric on our extension

Classifier	Accuracy	Precision	Recall
Random Forest	99.3%	99.7%	99.3%
SVM	98.8%	99.3%	98.3%
KNN	98.5%	97.01%	100%

Based on the obtained results we use the Random forest as a classification technique.

4.6 Testing Scenarios

In this section we will test our extension to detect unknown sites that belong to four classes: malicious sites, covid themed sites, benign sites, covid benign sites:

4.6.1. Testing with unknown malicious sites

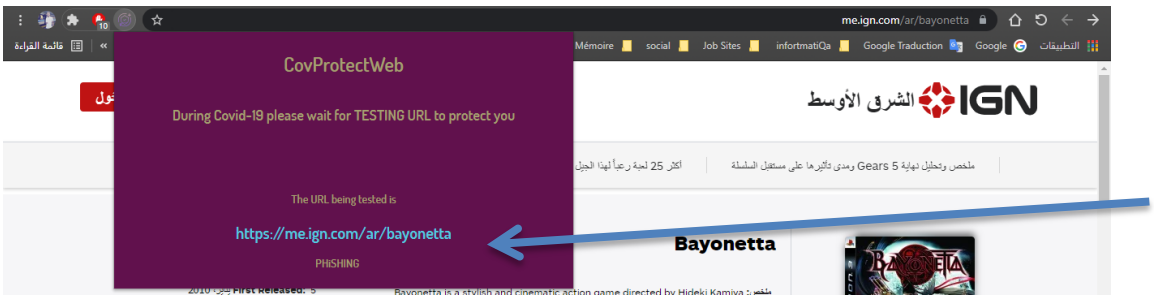


Figure 4.13: CovProtectWeb detect malicious sites unknown.

4.6.2 Testing with malicious site themed by covid-19

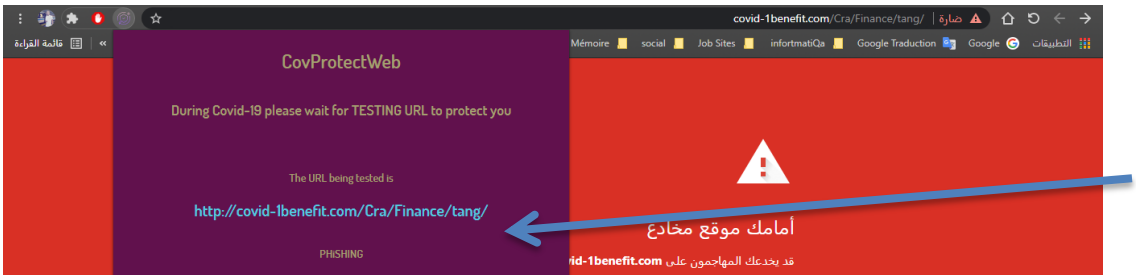


Figure 4.14: CovProtectWeb detect malicious website phishing themed Covid-19.

4.6.3 Testing with benign site

Figure 4.15: CovProtectWeb detect benign website.

4.6.4 Testing with benign site themed by covid-19

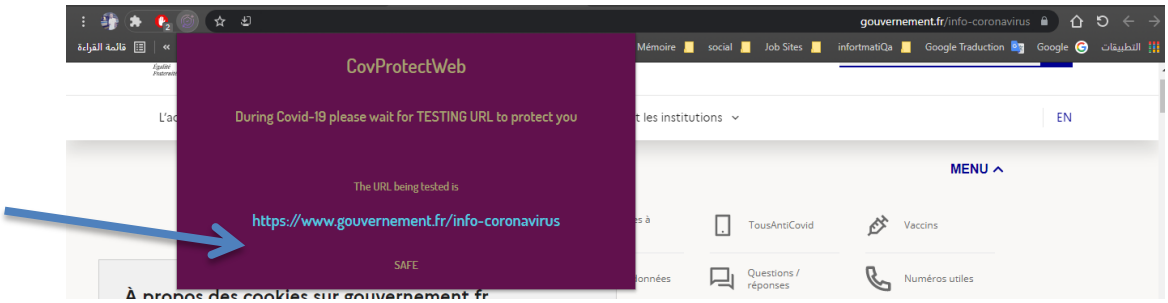


Figure 4.16: CovProtectWeb detect benign website themed by covid-19.

After testing all possible scenarios: known or unknown sites, themed COVID or no , malicious sites or no , we conclude that our extension can detect the majority of malicious content in general and especially themed COVID, because themed COVID malicious content has the same features as malicious content with its own WHOIS and browsing features.

4.7 Comparative study

The chart in figure 4.17 compare our classification evaluation metrics to the closest approach from earlier studies [14] (Approach 2) by evaluating the same dataset on Approach2 and using the common classifiers between them.

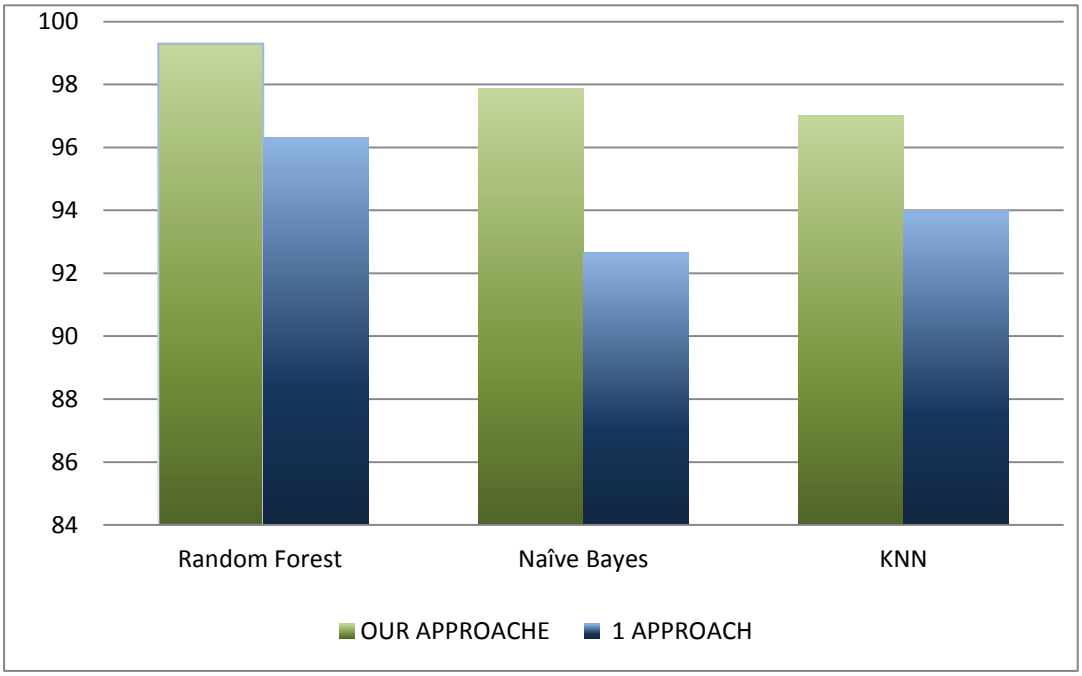


Figure 4.17: Accuracy comparison

4.8 Result discussion

During the test, our extension received excellent accuracy scores, which are the consequence of a well-chosen feature set that enhanced the results and turned our extension into a potent tool against web malicious content themed COVID-19, while approach 2 is limited to just 5 features: length of domain name , number of hyphens, Number of numeric characters ,shannon entropy of the domain name and Binary rating of URL.the three first features are features of general malicious content.

4.9 Conclusion

In this chapter, we introduced and tested our CovProtectWeb extension to detect malicious web content related to the COVID-19 virus or content affected during the COVID-19 pandemic. The way we categorize extensions is based on a specific feature set.

The pilot study clearly demonstrated the efficiency of our extension in detecting harmful covid -19 content, which was demonstrated by the high results obtained from the random forest classifier.

GENERAL CONCLUSION

General Conclusion

Because of the broad usage of the Internet at the time of the Covid-19 virus's spread, dangerous web content is spreading widely and quickly throughout global websites, particularly those carrying fake news, putting users' data at risk of being lost or exposed through various sorts of cyber-attacks. In our project, we're looking into the current methods for detecting dangerous web content, particularly that which is infected with the COVID-19 virus. The study revealed that these methods are still insufficient and have numerous limitations, as well as the fact that they are only studies and not an online addition, which drove us to create the Google Chrome extension "CovProtectWeb" to detect and fight harmful content the COVID-19 themed on the web by monitoring requests and responses from websites.

The following points were maintained by our work:

• List of new malicious characteristics related to COVID-19 web content.
• The important point in our project is the generation of malicious covid dataset from scratch that includes malicious and benign websites which weren't' used in previous works.

• Convert literary studies into an extension.
• Give descriptions on the topic of Covid-19 cyber-attack.
• Pretending to have found the majority of damaging content on the COVID-19 subject.
• Protect users from malicious content in general & precisely COVID-19 related content by blocking & notifying.
• Development of a machine learning model that can classify COVID-19-themed webpages into nearly flawless hazardous and benign categories.

During the experiments we tested our extension with the generated dataset (400 URLs 100 secure Alexa top Sites and 100 secure sites covid-19 it also contains 100 dangerous sites that carry the harmful word COVID-19, And 100 site are generally dangerous.)
The obtained experimental results demonstrate that our extension detects perfectly the majority of malicious websites themed covid-19 or not during the test phase, which prove the effectiveness of our selected features list to detect malicious content.

Perspectives

Our approach focused on studying lexical URL characterization; analyzing the request more than the response, and could be improved by focusing more on analyzing response pages, and adding a database with user feedback to improve the ranking process.

REFERENCES

References

Articles

[05]: WANG, P. ALI, A. KELLY, W. “Data security and threat modeling for smart city infrastructure “, Industrial Control System and Communications (SSIC), Shanghai, 2015.

[07]: Hallaraker, O. and Vigna, G. “Detecting malicious JavaScript code in mozilla,” in Proceedings of the 10th IEEE International Conference of Engineering of Complex Computer System, pp. 85–94. 2005

[08]: Eshete, B. “Effective analysis, characterization, and detection of malicious web page,” in Proceedings of the 22nd International Conference on World Wide Web companion. International World Wide Web Conferences Steering Committee, pp. 355–360. 2013

[09]: Gerardo, C. Mercaldo, F. and Visaggio. C. A. "Malicious javascript detection by features extraction." e-Informatica Software Engineering Journal 8.1 (2014).

[12]: Pritom, M. M. A. Schweitzer, K. M. et al. “Data-Driven Characterization and Detection of COVID-19 Themed Malicious Websites.” IEEE.In 2020 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 1-6). November 2020

[13]: Cordey, S. “The Evolving Cyber Threat Landscape during the Coronavirus Crisis”2020.

[14]: Ispahany, J. & Islam, R. “Detecting Malicious URLs of COVID-19 Pandemic using ML technologies”. arXiv preprint arXiv:(2009.09224). 2020.

[25]: Haoud, M. Djehiche, R. Saoudi, L. ‘Web Guard: Google Chrome Extension for Malicious Web Content Detection’. In Trends and Applications in Information Systems and Technologies. WorldCIST. Advances in Intelligent Systems and Computing, vol 1365. Springer, Cham. 2021

[27] Moraes, R. Valiati, J. F. et al “Document-level sentiment classification: An empirical comparison between SVM and ANN”, Expert Systems with Applications, 2013, pp 621-633

[28]: BREIMAN, Leo. Random forests. Machine learning,vol. 45, no 1, p. 5-32. 2001

[34]: Brownlee, J. “Machine learning mastery with python”. Machine Learning Mastery Pty Ltd, 100-120, 2016

[36] Vargiu, E. & Urru, “MExploiting web scraping in a collaborative filtering-based approach to web advertising”. Artif. Intell. Research, pp, 44-54. 2013

References

Web sites

[01]:XIAO Y., FAN Z., « 10 tendances technologiques à surveiller pendant la pandémie de COVID-19»https://fr.weforum.org/agenda/2020/05/10-tendances-technologiques-a-surveiller-pendant-la-pandemie-de-covid-19/. visited on juin 2021.

[02]:BEST I., « Paiements: les chiffres précis de la Banque de France sur l'effet du confinement » https://pointbanque.fr/2020/06/26/paiements-les-chiffres-precis-de-la-banque-de-france-sur-leffet-du-confinement/. visited on juin 2021.

[03]:JONES D., « Cyber fraud surges as COVID-19 changes banking, e-commerce » https://www.mobilepaymentstoday.com/articles/cyber-fraud-surges-as-covid-19-changes-banking-e-commerce. visited on juin 2021.

[04]:https://gatefy.com/blog/what-malicious-url/ visited on May 2021

[06]:https://patchstack.com/what-is-seo-spam-cloaking/ visited on juin 2021

[10]:https://gatefy.com/blog/what-is-phishing/ visited on May 2021

[11]:Michael Aliperti, "What You Need to Know About COVID-19 Scams," https://www.cisecurity.org/newsletter/what-you-need-to-know-about-covid-19-scam/APril visited on juin 2021.

[15]:https://www.trendmicro.com/vinfo/gb/security/news/cybercrime-and-digital-threats/coronavirus-used-in-spam-malware-file-names-and-malicious-domains visited on juin 2021

[16]:https://searchnetworking.techtarget.com/definition/URL visited on juin 2021

[17]:https://developer.mozilla.org/fr/docs/Learn/Common_questions/What_is_a_URL visited on juin 2021

[18]:https://searchnetworking.techtarget.com/definition/URL visited on juin 2021

[19]:https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL visited on juin 2021

[20]:https://www.thebalanceeveryday.com/what-does-url-mean-897078 visited on juin 2021

[21]:https://www.howtogeek.com/181767/htg-explains-what-is-https-and-why-should-i-care/ visited on juin 2021

[22]:https://www.dummies.com/programming/networking/url-manipulation-hacks-in-web-applications/ visited on juin 2021

[23]: https://checkphish.ai Url Scanner to Detect Phishing in Real-time | CheckPhish visited on April 2021

[24]:Free COVID-19 Threat List - Domain Risk Assessments for Coronavirus Threats (domaintools.com) visited on April 2021

[26]:Google Safe Browsing API/ Available at: https://developers.google.com/safe-browsing. Visited on juin 2021

[30]:https://gs.statcounter.com/ visited on juin 2021

[31]:https://nira.com/chrome-developer-tools/ visited on juin 2021

[32]:https://www.tutorialspoint.com/php/php_environment.htm visited on juin 2021

[33]:https://www.codementor.io/@benjamincohen/intro-to-machine-learning-nlp-with-python-and-weka-argnk39jr visited on juin 2021

[35]:https://www.pluralsight.com/guides/web-scraping-with-beautiful-soup visited on May 2021

[37]:https://brandastic.com/blog/what-is-alexa-rank/ visited on juin 2021

[38]:https://www.baeldung.com/cs/levenshtein-distance-computation visited on juin 2021

[39]:https://www.oreilly.com/library/view/mastering-python-for/9781788992510/2c8c5da0-5294-47ed-9cda-48fd31816191.xhtml visited on juin 2021

[40]:https://pypi.org/project/python-whois/ visited on juin 2021

[41]:https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/weka-gui-learn-machine-learning/ visited on Juin 2021

[42]:https://www.sciencedirect.com/topics/computer-science/classification-performance visited on juin 2021

Abstract

In light of the spread of the Covid-19 virus and the necessity of using the Internet as remote work, malicious content on the web has become a global threat to web users, as its huge spread has made users vulnerable to all kinds of cyber-attacks that can be carried out behind websites. Several recent researches have been proposed to detect harmful content on the subject of COVID-19, however detection of COVID-19 themed malicious websites has not been entirely resolved until now.

In this work, we propose a chrome extension to detect malicious web content with the theme of COVID-19 by analyzing HTTP requests and responses. The extension relies on a list of malicious features built with a machine learning classifier, and it does more protection. The experimental results obtained demonstrate the effectiveness of our extension in detecting malicious content with the theme of COVID-19, which was demonstrated by the perfect accuracy score.

Keywords: COVID-19, Malicious Websites, Web Browser Extension, Detection, Machine learning.

ملخص

في ظل انتشار فيروس كوفيد١٩ وضرورة استخدام الانترنت كالعمل عن بعد أصبح المحتوى الضار على الويب خطرًا عالميًا على مستخدمي الويب ، حيث أدى انتشاره الضخم إلى جعل المستخدمين عرضة لجميع أنواع الهجمات الإلكترونية التي يمكن تنفيذها خلف مواقع الويب. العديد من الأبحاث الحديثة قامت باقتراح حلول للكشف عن المحتوى الضار الذي يحمل موضوع كوفيد١٩ لكن هاته الأعمال لم تحل المشكل بشكل جدري.

.في هذا العمل ، نقترح امتداد chrome للكشف على محتوى الويب الضار الذي يحمل موضوع كوفيد١٩ من خلال تحليل طلبات واستجابات HTTP. يعتمد الامتداد على قائمة من الميزات الضارة المصممة بمصنف التعلم الآلي ، كما أنه يقوم بالمزيد من الحماية . توضح النتائج التجريبية التي تم الحصول عليها فعالية امتدادنا في الكشف عن المحتوى الضار الذي يحمل كموضوع كوفيد١٩ والذي أثبتته دقة النتيجة المثالية المسجلة على عدة فئات .

الكلمات المفتاحية :كوفيد-19 ، مواقع ضارة ، امتداد متصفح الويب ،الكشف، التعلم الآلي .