Machine learning as a complement to Https to prevent phishing attack on web site.

Lionel Sedar Noutegne[1] and Professeur Djiby SOW[2]

[1] African Institute for Mathematical Sciences, Km2 route de Joal (Centre IRD), 1418
Mbour-Thies, Senegal
noutegnelionel@gmail.com, noutegne.t.l.sedar@aims-senegal.org

[2] Training and Research Group "Discrete Mathematics and Cybersecurity",
Department of Mathematics and Informatics Faculty of Science and Technology
Cheikh Anta Diop University 5005 Dakar, Senegal
sowdjibab@yahoo.fr

Abstract. The presence of Https green padlock at the right end of the web browser Url bar of a website would provide users with tangible proof that their informations shared on this web site are well confidential. It would seem that Https functionality are nowadays strongly threatened by cyber attacks that use this characteristic to abuse users trust. Https would no longer guarantee full confidence of data traffic on web pages at a time when even phishing sites also have this secure socket layer on their website, with certificates that they signed themselves, thus putting users in total confusion. In this paper, we study the potential for using Visual identity, Content variation similarity, some intelligent rules on Url of a website to which we combine Machine Learning techniques to help Https better discriminate legitimate websites to phishing ones. Our algorithm spots phishing sites with an accuracy ranging of 92% to 96%.

Keywords: Machine learning, Cryptography, Cyber security, SSL/TLS, Https, Phishing, Content variation similarity.

1 Introduction

Recent web-based services such as those related to e-commerce, banking and many others handle confidential user data, and therefore require websites with high and guaranteed levels of security to avoid negative consequences for both website owners and their customers. Https has been announced for some years as meeting this level of security. It seems that Https features are nowadays strongly threatened by cyber-attacks that use its appearance to disrupt users' awareness. In term of the sensitivity of the data circulating on web pages, we are witnessing the emergence of various forms of lusts and in particular those associated with criminals web attacks who are constantly multiplying strategies to steal and abuse the trust of users' information. The majority of existing websites are in the form of hypertext transfer protocol in short Http but more importantly in its evolved version Https. It seems that the Https, which is ensure the confidentiality of user data passing through the forms on the web pages, is seriously threatened by a growing number of criminal cyber attacks. All this is confusing about the real quality of websites with Https. Faced with this threat, the major challenge is to propose new methods to help Https by panting at users of all forms of websites who try to abuse the trust web users. It is therefore for this reason, we use Machine learning technique and hash function concepts (Visual perception and Content variations similarity on websites pages) but also the intelligent rules on associated websites Urls, most of which are available in the form of phishing emails. Phishing is a form of cyber-attack and even the most powerful since 2018 [1] whose purpose is to recover any type of information that users of websites would provide through web pages form without the user being aware. The consequence is that the person in charge of this attack can access your confidential without your permission. Phishing nowadays affects large companies like Facebook, Amazon, Banks, Micro-finances, E-commerce websites and many others. The situation is such that the hacker sends you an email that you can receive via an email application and which contains a suspicious link. By trying to click on this link, you will get the first impression that you are on your banks websites because this is 95% true if you want to focus on the website Visual aspect. The other 5% would reside on the Url information and on the Content of you website banked provider. Statistics show that 90% of web users dont take time to check the web pages they are visiting.

They are so confident in the green ”Https security” padlock attached to the websites they are always visiting and it’s even a proof to them that they are on a very secure website. What they know is that the website has a certificate and therefore any information they provide would not be subject to any threat of protection and that’s why the padlock is green or grey in some cases. What they don’t know however, is that the phisher also simulate Https websites by signing SSL/TLS certificates and put them on phishing websites. In such situations, it will then become very difficult to distinguish what a legitimate website from a pirate one for these users. This problem has led to a lot of work and publications, but we are still looking for new and more efficient techniques. To our knowledge, Visual perception and Content variation techniques (and more broadly image and text hash functions), associated to intelligent rules on Url of web pages, combine to machine learning are rarely used explicitly to detect fraudulent websites on the Web. Weifeng and Xia [2] and Fu et al [3], respectively, use Phash and EMD algorithms to search between legitimate websites and those seeking to abuse users. Mensah et al [4], on the other hand, analyze the Visual aspect within web pages to create a similarity between it for the search of relevant information and distinguish what a legitimate page look with phishing one. Their approach is based on the dHash algorithm for Visual perception. In this paper, we try to show the ability to use the combination of several different detection approaches, namely Visual perception techniques as well as content variation similarity within a web page, features related to the Url of a website, all coupled with machine learning techniques to discriminate web pages as legitimate or Phishing. The rest of this document is organized as follows: In

Section 2, we present related work on other methods to come against phishing. In section 3, we describe the Perceptual hash technique, the Content variation similarity approach based on the Sha3 function of web page and the analysis of the different rules on associated Url. Section 4 is devoted to evaluating the proposed methodology for our study, and the result of its application to a data set that we generate automatically through our ingenuity, and then we present the results obtained. In section 5 we present the discussion. Finally, in section 6, we conclude our work and provide perspectives

2 Related work

Since, the Phisher generally uses wrote programs for a specific website in order to mislead users and retrieve information, we have nevertheless encountered in the literature techniques developed to fight against this identity theft. Anti-phishing techniques are divided into two main families: technical and non-technical approaches.

2.1 Anti-phishing list and rule interfaces

Many systems implement black and white list concepts to Wrestling phishing attacks. The Blacklist method (Prakash et al [5]) uses a phishing or domain Url blacklist to block phishing Urls. It's only effective in detecting known phishing Urls. A blacklist is usually based on time-consuming human feedback that is ineffective in blocking phishing on short term web pages. Blacklists can be obtained from sites such as PhishTank (www.phishtank.com) and Netcraft (toolbar.netcraft.com). The black list is implemented on most web browsers. The white list, on the other hand, aims to identify good known sites by maintaining a white list of approved sites in Url or domains. Any Url not included in the white list will be blocked. White lists are also implemented on most web browsers.

The most of the existing method is based on the Url syntax. It consists in defining a certain number of rules on the Url based on the RFC 1738 proposed by the founding father of the Web Berners-Lee [6]. Following this logic, BENAMMAR [7] proposed an approach that is essentially based on the characterization of Url as defined in RFC 1738. Fette et al [8] have succeeded in proposing system to detect phishing emails with a set of features they define on website domain such as the age of the domain, the presence of a certain number of suspicious characters in the Url that TimBerners-Lee considers suspicious characters and therefore should not appear in a Url website. The disadvantage of this method is its simplicity because it does not integrate malicious content analysis.

2.2 Automated phishing similarity detection

The weakness of the black and white lists is they don't have phishing detectors based on anomalies, which are based on classification model with discriminatory rules. The classification model can be built prior from knowledge. The DOM or Document Object Model is a cross-platform, language-independent convention for demonstrating objects in XML, XHtml or Html documents. The DOM is a tree representation. In the DOM-based phishing detection system, the DOM tree of the suspicious web page is compared with legitimate web page in order to create a matching, i.e. a kind of pattern similarity of the zone structure to be compared. Nguyen et al [9] there very well demonstrated work on phishing websites using the DOM. Hara et al [10] proposed a phishing detection technique that stores an image database to compare and determine the type of the website. The image database contains images and the corresponding domains of legitimate websites and phishing. From a given site image the tool is likely to tell if it's legitimate website or not.

2.3 A human factor as strong link in phishing detection

Phishing pages are sent most of time by email. In general, some elements may allow you to recognize phishing email. Phishing messages are almost authoritatively, and it's important to never give up to these underlying threats. It's expressions such as: " *your account will be completely deleted* "; the promises " *you would get a refund of X dollars" or the guarantees " we take our customer private life very seriously*" etc... Statistics shown that in general 98% of users don't take time to check the Url of pages that the try to fill in their confidential information. Yet, it's one of the things to do beforehand. In general, these emails contain a link that can be clicked whose first reading may seem trustworthy, but which in reality redirects to a fraudulent website. The phishing domain names or the address server to which they claim to be attached are such that if you check carefully, you will find erroneous syntax and punctuation, lack of accent, mixture of English and French words or other language, formatting errors. Non-technical methods to fight phishing increasingly include the human factor. Users don't still realize their importance in this exercise, especially since behaviours don't evolve as quickly as technical systems would like. Cyber-attackers are well aware of this lack of user knowledge and are systematically targeting as part of massive phishing operations with the aim of getting victims to validate links before the technical systems have time to react. Non-technical methods are therefore more in line with awareness and recommend a number of tips to users.

It is important to remember that in general your supplier will never ask you to provide information such as credit card numbers or passwords by e-mail since they already have them for most of time. In addition, don't be convinced or intimidated by the threats of the underlying messages. A number of good practices allow us to protect ourselves against any phishing attack: It is recommended at first to never answer an email or click on any links contained in an email and which appear suspicious. Or to check that the address of a secure website begins with "**Https://**" even if it's not a sufficient condition to guarantee the legitimacy of a website. It's recommended to avoid filling in forms requesting your confidential information Arachchilage et al [11] and Hale et al [12], have each oriented themselves towards a game-based approach. They place the user in a fictitious environment where they are confronted with use cases intended to be close to reality and to identify whether a web page is legitimate or not. In this process, the objective is to instill in users the habit of always checking the settings before trusting website. The famous adage ”don’t believe in everything you see” is the watchword to follow to avoid being caught in the trap by a phishing email. Most of the time, the emails we receive come from people we know fully, it can be an email with contacts from a relative, a private individual or our organization and therefore sufficient to rush to the email and open it. It’s recommended don’t rely on the display name of the email sender. Don’t we say that prevention is better than cure, we mean here if something seems unclear or strange, it is better to don’t even open the email for fear of being disappointed afterwards, and it is also highly recommended to be careful with attachments. Phishers, in the logic of deceiving the vigilance of their victims, will often attach false icons and images that are not in reality that of your bank or your organization and so on. The major challenge in the coming years will be to raise awareness among users through various training actions and Training section in order to change their daily behaviour. Information systems security managers must therefore seek today to transform the user into a proactive member of security for greater protection.

3 Hash functions and intelligent rules approaches for detecting phishing websites

We try to study here on the basis of features (cf.fig.1) such that the intelligent rules applied to the Url syntax of a website and associated with his content, the Visual perception and the similarity in the variation of the Content on pages of the website to which this Url is attached. Then we apply machine learning techniques to assess the legitimacy of the pages of website. This is actually a task that would be very difficult for users who are victims of sensitive information theft. The solution is therefore of great importance for web browsers, especially those who already implement SSL/TLS security. The first two concepts we introduce in this paper for phishing detection are the concepts of Visual perception and Content variation similarity of web page. It’s important to remember that it will be relatively difficult for a user to identify at first sight, the Visual and variation difference between phishing and its legitimate version. So you shouldn’t totally trust web pages that appear before you asking you for a certain information, even if it’s your bank’s website. The proposed approach is new because we adapt the concept of hashing function defined on text of web page content using the Sha3 hash function. On the other hand, we also use the concept of Visual perception, which is also based on images hash function. The technique used is different from that proposed by [13], or by [3], but is based on the dHash algorithm which was used by [4]. The difference from the proposition by [4] is the way we evaluate the depth of the pages of the website to be compared using their fingerprint defined by dHash. We open the pages of given website randomly and sometimes on three levels. As long as links to other pages are randomly taken from each other after opening them. The Sha3 hash function introduced in this paper helps us to evaluate the similarity in terms of variation between the pages of the opened websites. This feature is significant for phishing websites, where many of which are single content page requiring the victim to fill in confidential information. These different authentication techniques have emerged to verify the integrity of the content and prevent forgery. By using Hamming distance (HD), it will be possible to evaluate two confusing files or images at near-bit level.

1. **Content Variation Similarity**

The concept of Similarity that we are referring here is based on the Sha3 function. If we consider two web pages as two text files present on a website, one way to check if these pages pages are different, would be to evaluate their respective fingerprint. Hash functions like Sha3 are very well suited for such tasks, in that from a certain threshold of the Hamming distance between digital hash value of these two files, we can say exactly if the are some variations between them, in most cases, any change in bit would confirm the level of divergence that exists between the two web pages. The hash function is based on the Sponge construction of [14]. This architecture includes an Absorption Phase and a Pressing or Exit Phase called Squeezing Phase. The Sha3 hash function starts with:

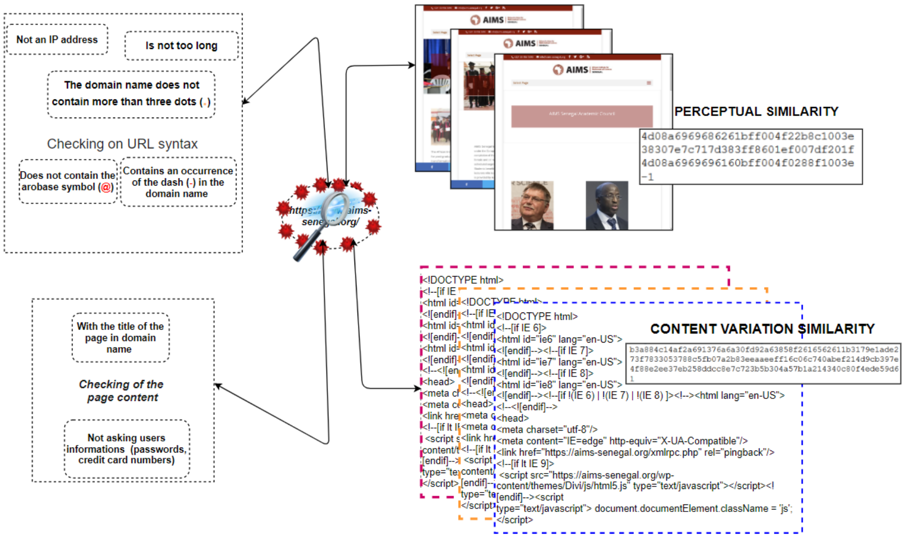
- The extension of the message (file) to length of 1600 bits on the input,
- Through a permutation set of the Keccak function [14] over a number of 24 rounds
- On outputs a string of fixed lengths or associated fingerprint

2. **Visual perception**

The Visual perception to which we refer in this paper defines the Visual representation that a user-side web page reflects. The Sha3 hash function above may not meet the integrity and authentication requirements for this type of image. Visually, to be able to modify an image, it's important that the modification is as deep as possible so the Sha3 hash functions are sensitive to a single bit change in the fingerprint in associated image. For our approach, to identify two pages as belonging to the same domain, we use the dHash [15] perceptual hash function to obtain the image hash and we define a threshold of difference in the Hamming distance of the fingerprints associated with these images. The algorithm is done as follows:

- Convert the image into a gray scale image and calculate difference between two adjacent pixels
- Assignment of bits according to whether or not the left pixel is a brighter pixel than the right one
- At the output is then a hexadecimal string representing the hash of the image.

We will be able to realize that it's also possible to retrieve an web page document as a screenshot with his ergonomics contains, and obtain all the text content of the same page. So, when we introduce the concepts of hash functions (Different hash on images and Sha3 on text), we can then have the hashed versions of the page screenshot but also the text content. Considering these features, the system will be able to perform analyses based on intelligent rules defined on Url of a website.


Not an IP address
Is not too long
The domain name does not contain more than three dots (.)
Checking on URL syntax
Does not contain the arobase symbol (@)
Contains an occurrence of the dash (-) in the domain name
With the title of the page in domain name
Checking of the page content
Not asking users informations (passwords, credit card numbers)
PERCEPTUAL SIMILARITY
CONTENT VARIATION SIMILARITY

```
4d08a6969686261bff004f22b8c1003e
38307e7c717d383ff8601ef007df201f
4d08a6969696160bff004f0288f1003e
-1
```

```
<!DOCTYPE html>
<!--[if IE 6]>
<html id="ie6" lang="en-US">
<![endif]--><!--[if IE 7]>
<html id="ie7" lang="en-US">
<![endif]--><!--[if IE 8]>
<html id="ie8" lang="en-US">
<![endif]--><!--[if !(IE 6) | !(IE 7) | !(IE 8) ]><!--><html lang="en-US">
<!--<![endif]-->
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<link href="https://aims-senegal.org/xmlrpc.php" rel="pingback"/>
<!--[if lt IE 9]>
 <script src="https://aims-senegal.org/wp-
content/themes/Divi/js/html5.js" type="text/javascript"></script><!
[endif]--><script
type="text/javascript"> document.documentElement.className = 'js';
</script>
```

```
b3a884c14af2a691376a6a30fd92a63858f2616562611b3179e1ade2
73f7833053788c5fb07a2b83eeaaeeff16c06c740abef214d9cb397e
4f88e2ee37eb258ddcc8e7c723b5b304a57b1a214340c80f4ede59d6
1
```

Fig. 1: Modeling the check view of the different features analysis.

3.1 Url analysis rules

The purpose of this section is to present a set of rules developed on the fundamental features of the Url and type of content presented by websites. Most of these features have been found in the literature review. We implemented each feature to return either -1 if it looks like a phishing page, or +1 if it doesn't. We present in the following 7 rules which we considered important for our discrimination system.

3. **Use of the symbol "(@)" in the Url : Feature 3**

This feature checks if the Url of a page contains the symbol "(@)". A "(@)" in a Url ensures that the character string on the left is ignored, so the character string on the right will be the string recognized by the browser. This heuristic was also used in [8].

4. **The Name of the resource with the "(.)" : Feature 4**

Suppose we have the following link http://www.aims-senegal.sn.org/portal.com. A domain name always includes the top-level domain or top-level suffix. We note that legitimate links have two points in the Url since the "**www.**" entry can be ignored. For our study, we try to generalize these different scenarios, by taking an interest only to resource name. However, if the number of points is greater than three, the Url is classified as could be phishing. because it will have several sub-domains. This heuristic was also used by [8].

5. **IP address as domain name : Feature 5**

In Web jargon, an IP address is a unique number assigned to each device that identifies it on the internet. The $IPV4$ is encoded on 32 bits and is composed of 4 digits from 0 to 255, separated by dots (YY.YY.YY.YY.YY.YY). An IP instead of the domain name would give us a dissuasion and for that, we consider a Url with a IP in the domain name as could be phishing. This heuristic was also used by [4].

6. **The length of the Url : Feature 6**

It is used to hide the suspicious part in the address bar. There is no reliable length data to distinguish Url because it varies from browser to another. However, we note that phishing pages generally have a very long Url.

7. **Presence of the "$-$" character in the Url : Feature 7**

Infrequently used in domain names of legitimate websites, the special character "$-$" dash in the domain name give the user impression to make him believe that he is on a legitimate page. We consider a Url with more than three occurrences of this character in the domain name or with more than four occurrences in the sub-domain name as could be phishing. page.

8. **Form requesting credit card or password: Feature 8**

We consider pages forms that asking confidential information as phishing. We automatically identify a page without a form as a secure. This heuristic was also used by [13].

9. **Url with title in domain name : Feature 9**

Secure sites generally put domain name in the $< title >< /title >$ tag of the Html code. Phishing pages tend to imitate this characterization by adopting syntax close to those of the sites they are trying to usurp but never exactly the same domain name as the legitimate sites. We automatically identify a page that does not comply with this convention as could be phishing. This heuristic was also used by [4].

4 General architecture and solution evaluation

4.1 Overview

The system we propose in this paper is based on the architecture of Fig.3, and in this section we want to describe in detail how discriminated legitimate websites for those who try to abuse the trust of web users. Through our studies and observations, we have realized that legitimate websites move very little and not on any other website, since we know that there is a SSL/TLS certificate relationship between websites that are legitimate and those that are not. It will be difficult for a legitimate World Bank website for example to allow malicious websites to access its pages. In such cases, we proceed as follows: we successively open the pages of the sites at three levels Phishing pages are such that, attackers sometimes omit any links that can lead the victim to investigate the malicious content. These are pages with very little or almost no link but a form asking to fill sensitive information. In this case, our analysis returns almost Visual and variation Content similarity close to zero and it would mean that the malicious content probably didn't changed. Obviously, those features are not significant for legitimate website that generally have several pages in their domain. Moreover, the score of the Visual perception of the pages of legitimate websites is not very high because its pages tend to remain in the same domain with almost identical Visual similarity on all pages, the algorithm is rolled out on our database of phishing and legitimate websites.

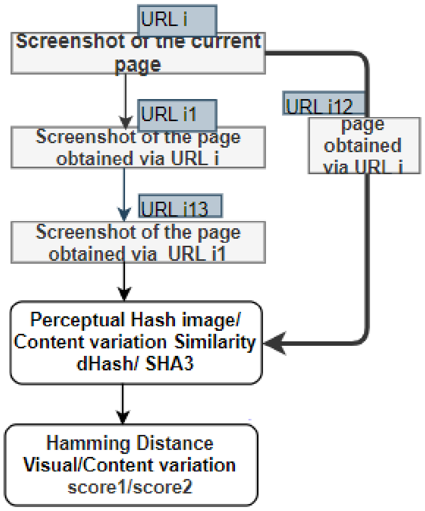

URL i
Screenshot of the current page
URL i1
Screenshot of the page obtained via URL i
URL i13
Screenshot of the page obtained via URL i1
URL i12
page obtained via URL i
Perceptual Hash image/ Content variation Similarity dHash/ SHA3
Hamming Distance Visual/Content variation score1/score2

Fig. 2: Process of similarity scores based on the Visual and textual aspect of a web page.

When we choose a Url that we note here by **Urli**, we open it in the browser (for our work, we used the FireFox ESR browser). The page being opened in the browser, we take a screenshot of it and we randomly select two links on the page that we use **Urli1** and **Urli2** and the Chrome Webdriver closes for a while. Then, with BeautifulSoup, the page content from Urli, Urli1, Urli2 is retrieved. At this level, the Chrome Webdrive reappears again, opens Urli1, takes the screenshot associated with this Url, and a link is randomly selected on the page named by Urli3 and from this point, Webdrive closing Webdrive for a short period of time. Following this, there is also the page content of Urli1 being recovered, the Webdrive appears for a third time and also generates the screenshot of the page associated with Urli3. The Fig.2 illustrates an overview of the process. Adding that after each Webdrive closure, the screenshot of Urli, Urli1 and Urli3 are transmitted into the dHash algorithm to generate their respective hash values. Each time content is retrieved (Urli, Urli1 and Urli3) by BeautifulSoup, the different results are passed to the text hash function (Sha3). We note respectively by **MUrli**, **MUrli1**, **MUrli2** the results of the respective Sha3 hash of Urli (our first page) and Urli1, Urli2 (the pages selected on Urli). Similarly, the results of the perceptual hash image of Urli, Urli1, Urli2 are respectively referred to as **dUrli**, **dUrli1**, **dUrli3**. We obtain by using the Hamming distance (DH), the **scores2** and **score1** for the Content variation similarity and Visual similarity respectively and which are defined by the two relationships below:

$$score1 = DH(dUrli1 - dUrli3) - DH(dUrli3 - dUrli) \tag{1}$$

$$score2 = DH(MUrli1 - MUrli2) - DH(MUrli2 - MUrli) \tag{2}$$

Based on our database obtained via phishtank and Alexa, each phishing Url is labelled by -1 and legitimate one by 1. Each Urli and its sublinks (links obtained on its content) must go through the left modules to test their syntax and through the right module to test features defined on the content page (see Fig.3). At the output of each module, two types of decisions are issued considering all the criteria on the sequence of Urli, Urli1 and Urli2 and respecting the conditions defined in the left module (decision 1 which checks whether the conditions defined on the left-hand module are met at least two times out of three for each opening page). The process is applied similarly to have decision 2, but this time considering the right module. The output of these three components allows us to have a new database, containing Urls with scores on all the different features analysis and attesting whether or not they have respected the different assumptions defined. The new database being generated, it's now question for us to pass our data into the SVM algorithm in order to evaluate the level of discrimination.

4.2 Machine learning phase and result

In this section, we try to present the process by which the data were generated, as well as the classification results of our study data set. The algorithm in its entirety, has been implemented in python in its version 3.7 on a Laptop HP 15 Corei5 machine, 8th generation with 8GB of RAM memory and 1TB of hard

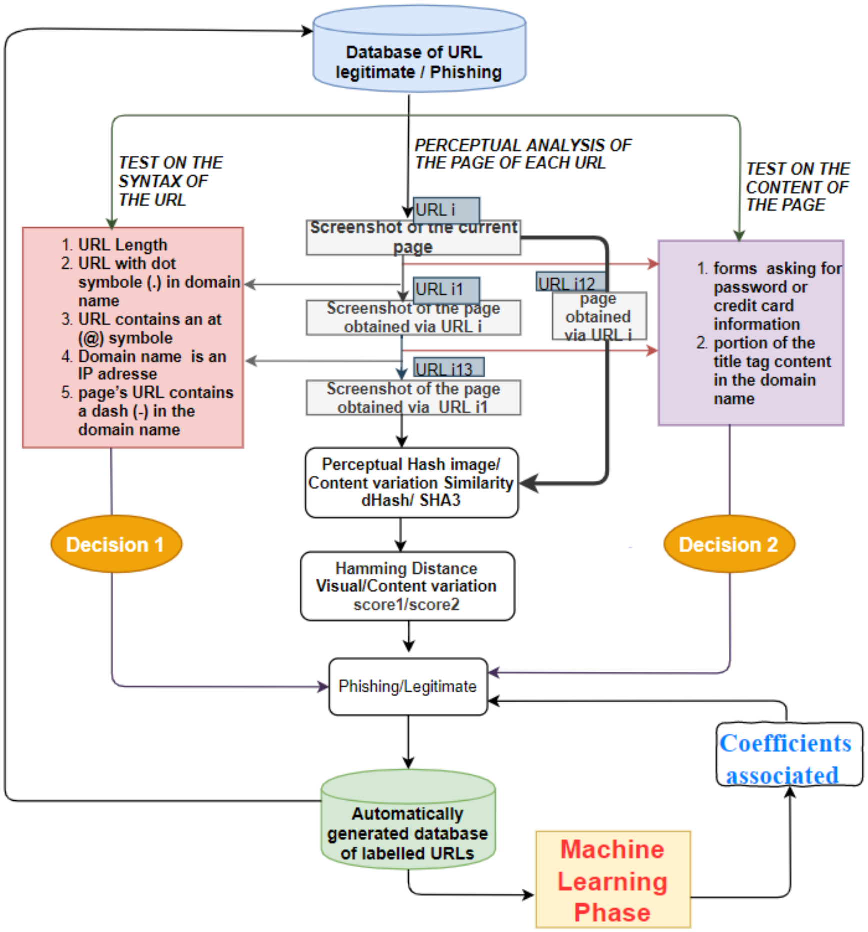

Database of URL legitimate / Phishing
TEST ON THE SYNTAX OF THE URL
PERCEPTUAL ANALYSIS OF THE PAGE OF EACH URL
TEST ON THE CONTENT OF THE PAGE
1. URL Length
2. URL with dot symbole (.) in domain name
3. URL contains an at (@) symbole
4. Domain name is an IP adresse
5. page's URL contains a dash (-) in the domain name
URL i
Screenshot of the current page
URL i1
Screenshot of the page obtained via URL i
URL i12
page obtained via URL i
URL i13
Screenshot of the page obtained via URL i1
1. forms asking for password or credit card information
2. portion of the title tag content in the domain name
Perceptual Hash image/ Content variation Similarity dHash/ SHA3
Decision 1
Decision 2
Hamming Distance Visual/Content variation score1/score2
Phishing/Legitimate
Coefficients associated
Automatically generated database of labelled URLs
Machine Learning Phase

Fig. 3: Proposed Solution Architecture.

disk. Data generation was carried out on 208 Url, 104 legitimate sites and the remaining 104 on phishing sites. It took us 7 hours and 30 minutes to generate data on all 208 websites. The splitting of the 208 sites was such that 156 websites were used for training and 52 websites for testing. The analysis of the 9 Features allowed us to perform some analyses, based on fourth sub-data sets of our large set, a question of what could be the contribution of the newly introduced concepts, respectively by Set1, Set2, Set3 and Set4 to assess the contribution of Visual perception, Content variation similarity, on intelligent rules and to evaluate the complete data set respectively. The results of the computation of each metric have been recorded in Table 1.

We can see that the number of false positives as well as the false negatives are less significant with the set4.

Table 1: Performance measurements on the Textual, Perceptual, Intelligent rules approach on Url and on the tree combine

Data sets	TPR	TNR	FNR	FPR	Precision	Recall	F1score	Error	Accuracy
Set1	0.62	0.64	0.35	0.37	0.64	0.65	0.67	0.38	0.63
Set2	0.64	0.70	0.29	0.35	0.71	0.63	0.67	0.36	0.67
Set3	0.85	0.95	0.04	0.17	0.96	0.81	0.88	0.15	0.92
Set4	0.96	0.96	0.04	0.04	0.96	0.96	0.96	0.04	0.96

This is also confirmed by the ROC curve (see Fig.4). With this ROC curve, we have an area under the average curve (AUROC) that varies by 0.63% with set1 defined for Visual perception to reach an average AUROC of 0.68% with set2 that is associated with Content variation similarity. The best performance is obtained with the fourth set of characteristics where the average AUROC is equal to 0.96%. Subsequently it was wise for us to confirm the performance choice of our classifier, we also launched our set4 on other machine learning algorithms and which are generally used for supervised learning problems. We used the Random Forest algorithm, logistic regression, the k nearest neighbour algorithm. We wanted to compare them to our SVM algorithm using an ROC curve. Fig.5 compares these different learning models with our SVM. In view of the results, we can say with great conviction that we would seem to have made the right choice on the algorithm model for our discrimination problem. The

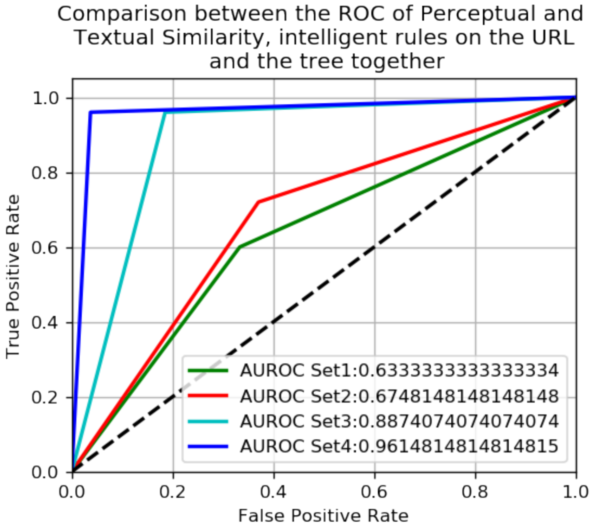


Fig. 4: ROC curve of the four sub-sets sous ensembles

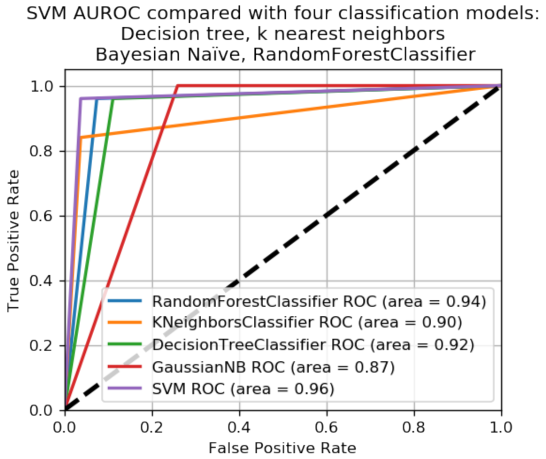


Fig. 5: Comparison Roc curve with other classification algorithms

SVM algorithm has a relatively larger area under the curve (AUROC) than the other classifiers.

5 Discussion

In this work, we presented the problem of phishing websites and the need to appropriate new techniques to come against this nebulous problem. More precisely on the use of Visual perception, Variation content approach within a website, combined with intelligent detection rules based on the syntax of a Url in a browser and on the consistency of the content to which it is associated. All coupled with a machine learning technique, to reinforce this trust around the Https web security protocol and its main layer SSL/TLS in the fight against phishing websites. It was noted that, given the sensitivity of the data circulating on the Web, hackers were multiplying strategies to recover user data to the point of impersonating their target's website by also offering certificates that were sometimes self-signed through the Https protocol, thus creating great confusion among users and their web providers about the legitimacy of a website. The approaches generally used to address this problem resulted in the use of less effective techniques cases are generally unique approaches.

Our approach being a hybrid solution, we presented every aspect of our solution starting with the concept of Visual perception and Content variation similarity within a website. This notion was as interesting as it allowed us to see the difference between the hashing of an image and the hashing of everything that is text within a web page as well as the different similarities associated with it to better assess the level of Visual identity between a phishing site and a legitimate site. Subsequently, we studied the intelligent rules on Url. In order to better understand what legitimate authentication between client and server respecting Tim Berners-Lee's methodology [6].

A machine learning algorithm was studied for this purpose discriminate. At this level, it was necessary to find the data adapted to our study, which was not easy, we could not find the data that met our requirements of this problem of discrimination. It was therefore question for us to build an automatic system for generating data from the various analyses. The model being well defined, we passed our data through our SVM algorithm and against all expectations, the accuracy of the model was evaluated at 96%. Our results are very appreciable compared some one see in the literature review. By building this solution, we face some difficulties. On the one hand, during data generation, it was very difficult to analyze some pages. Indeed, some websites almost don't allow scripts to access their content. Many sites have installed Captcha to prevent any unauthorized program from accessing their server, making it difficult to analyze, and given the difficulty in finding good data for learning problems with phishing websites. We plan to build a description of data set automatically generated for this work and make them available to the scientific community.

6 Conclusion

In this paper, we have presented the evaluation of a hybrid system for detecting phishing websites. An approach based on the combination of several different features, such as the Visual perception and Content variation similarity of a website pages, intelligent rules on the Url of a website. Our system is such that comparisons are made by opening the pages of websites up to a certain level. The question is to evaluate similarity of theses pages and the directions they can take. We described the implementation of our solution, and explained precisely the contribution of each feature in the construction and inefficiency of the overall system for obtaining good metrics. It was observed that our solution catches about 96% of phishing sites with about 4% false positives. As part of our future work, we plan to refine our solution to prepare for deployment and evaluation on a larger scale. We plan to synchronize our solution with the Https security installed on web browsers. We want the developed solution to be launched just after the SSL/TLS verification mechanism is completed. We will also continue to optimize our solution to improve its recall rate and reduce the false positive rate.

Acknowledgments

The work described in this document was supported by the African Institute of Mathematical Sciences of Senegal and the Cheikh Anta Diop University of Senegal.

References

1. Simon O'Dea, http://www.shortUrl.at/dvwIN. Last Criminal phishing trips: cyber attacks in Australia, 2018
2. Weifeng, ZG ZHANG and Xiantao, Tian: Detecting phishing web pages based on image perceptual hashing technology, IJACT: International Journal of Advancements in Computing Technology (2012),vol.4, pp. 139–145. https://doi.org/2
3. Fu, Anthony Y and Wenyin, Liu and Deng, Xiaotie: Detecting phishing web pages with Visual similarity assessment based on earth mover's distance (EMD), IEEE transactions on dependable and secure computing,vol.3, pp. 301–311,IEEE(2006)
4. Mensah, Pernelle and Blanc, Gregory and Okada, Kazuya and Miyamoto, Daisuke and Kadobayashi, Youki: AJNA: Anti-phishing JS-based Visual Analysis, to Mitigate Users' Excessive Trust in SSL/TLS, IEEE(2015) 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS),vol.3, pp. 74–84.
5. Prakash, Pawan and Kumar, Manish and Kompella, Ramana Rao and Gupta, Minaxi: Phishnet: predictive blacklisting to detect phishing attacks, Proceedings(2010) IEEE, pp. 1–5.
6. Berners-Lee, Tim.:RFC 1738: Uniform resource locators (Url), journal(1994):ftp://ftp. internic. net/rfc/rfc1738. txt.
7. BENAMMAR, Safi: Une approche a base d'Url pour la detection des sites Phishing.
8. Fette, Ian and Sadeh, Norman and Tomasic, Anthony: learning to detect phishing emails, Proceedings of the 16th international conference on World Wide Web(2007), pp. 649–656.
9. Nguyen, Le Dang and Le, Dac-Nhuong and Vinh, Le Trong: Detecting phishing web pages based on DOM-tree structure and graph matching algorithm, Proceedings of the Fifth Symposium on Information and Communication Technology, pp. 280–285, ACM(2014).
10. Hara, Masanori and Yamada, Akira and Miyake, Yutaka: Visual similarity-based phishing detection without victim site information, PSymposium on Computational Intelligence in Cyber Security, pp. 30–36, IEEE(2009).
11. Arachchilage, Nalin and Love, Steve and Scott, Michael: Designing a mobile game to teach conceptual knowledge of avoiding phishing attacks', International Journal for e-Learning Security, pp. 127–132,vol.2,No.1, Infonomics Society(2012).
12. Hale, Matthew L and Gamble, Rose F and Gamble, Philip: CyberPhishing: a game-based platform for phishing awareness testing, 2015 48th Hawaii International Conference on System Sciences, pp. 5260–5269,IEEE(2015).
13. Zhang, Yue and Hong, Jason I and Cranor, Lorrie F: Cantina: a content-based approach to detecting phishing web sites, Proceedings of the 16th international conference on World Wide Web, pp. 639–648,ACM(2007).
14. Paar, Christof and Pelzl, Jan, Xiaotie: SHA-3 and The Hash Function Keccak, Understanding Cryptography-A Textbook for Students and Practitioners, Springer Berlin Heidelberg(2010)
15. Viktor Popkov: Possible application of perceptual image, HTTPS://digi.lib.ttu.ee/i/file.php?DLID=2816&t=1, 2015.