A new approach for the detection and analysis of phishing in social networks : the case of Twitter

Kamel Ahsene djaballah
University Science and Technology
Houari Boumediene
Algiers, Algeria
adjab.kamel@gmail.com

Kamel Boukhalfa
University Science and Technology
Houari Boumediene
Algiers, Algeria
boukhalk@gmail.com

Zakaria Ghalem
University Science and Technology
Houari Boumediene
Algiers, Algeria
zakaria.ghalem95@gmail.com

Oussama Boukerma
University Science and Technology
Houari Boumediene
Algiers, Algeria
boukerma.oussama95@gmail.com

2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS) | 978-0-7381-1180-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/SNAMS52053.2020.9336572

Abstract— **Cybercriminals use Internet and social networks as a vector to launch phishing attacks in order to lure victims to disclose personal information. There are several methods for the detection and analysis of these attacks, among the most used are those based on machine learning. However, these methods suffer from a lack of precision in detecting phishing attacks. Therefore, it is necessary to propose new methods to improve the predictions of these attacks. In this article, we propone a three (03) step approach for the detection and analysis of phishing on Twitter, which can be applied to several other social networks. The first step is to browse a database called "Blacklist" in search of the suspicious URL (Uniform Resource Locator). Subsequently, we proceed to URL (Uniform Resource Locator) analysis leveraging machine learning techniques introducing new features. In this step the three following classifiers were used namely, Regression Logistics, SVM (Support vector Machine) and Random Forest. Afterwards, we added a module to analyze Twitter accounts, also based on machine learning, using user-related features to detect malicious twitters who are causing the phishing attacks. Finally, we tested our system on real data and then implemented it in the form of an application for the end users.**

Keywords— social networks, Twitter, Phishing, blacklist, analysis, machine learning, prediction.

I. Introduction

Phishing attack consist of the fraudulent attempt to obtain sensitive information such as login, passwords, and credit card number. It is one of the most pervasive cyberattacks on the internet, especially on social networks [21]. Cybercriminals use diverse techniques that cannot sometimes be detected by existing anti-phishing mechanisms. These include URL shortening, the use of subdomains and link manipulation techniques.

To detect phishing attacks, other than the verification in a phishing URLs database [4], there are three main approaches. The first one is based on the analysis of malicious site links or phishing URLs, either by using the features extracted from an URL [1], [10] and [3], or by using the Whois lookup service [6].

The second approach is based on content analysis of the web page or the text of the post in a social network (tweet for instance). The analysis of a phishing web page focuses on the properties extracted from the HTML code of a website to identify phishing [13], [5], [8]. Whereas the analysis of an email or a message on a social network (text of a tweet, publication, etc.) is based generally on natural language processing techniques (NLP) [6].

As for the third approach, a combination of the first two approaches: URL and content takes place [7], [9], [10], and [11].

To underline the specificities that social networks represent, on particular Twitter, where certain Twitters can share (retweet) malicious content, the researchers in [12] and [11] took into consideration the features specific to the account of the user posting the tweet, in order to determine the suspicious twitters.

The third approach presented above is the most effective method for detecting phishing as some URLs are stealthy and cannot be detected quickly, however the content of their corresponding web pages can help to extract other features that allow to classify a URL as phishing. However, this approach which uses machine learning techniques, despite its rather satisfactory results, can still be improved to curtail false positives. This makes the combination of the blacklisting approach and machine learning techniques very interesting to improve the accuracy of the solution.

In this article, we propose a new approach to detect phishing on Twitter. Our proposal combines the verification method in a database or "Blacklist" with the classification method by machine learning based on URL and web page content analysis. Furthermore, it integrates the analysis of Twitter accounts, to limit the sharing and spread of malicious phishing content (retweeted by Tweeters). In our view, none of the research studies cited have used the combination of these three methods. Besides, we used a combination of different features for machine learning, whether for URL analysis or user accounts.

The rest of this article is organized as follows; in section 2 we present some research work that have dealt with the problem of detecting phishing attacks on social networks.

Section 3 presents our approach for detecting phishing attacks on Twitter, as well as the complementary method for detecting malicious users.

Section 4 represents the implementation of our approach, followed by an evaluation and an interpretation of the results, tests, and demonstration of the developed application, before ending in section 5 with a conclusion and some perspectives.

II. Related Work

The detection of phishing, on the Internet and social networks has aroused the interest of many researchers. Note that contributions having treated this attack were based on either the use of blacklists methods [4], or classification methods, namely machine learning techniques to categorize URLs [14], [17] and [16], by analyzing the URL or the content of the web page [7], [9] and [10]. Moreover, some researchers have taken into consideration Twitter accounts features [12] and [11].

In [4], the authors tackled the problem of the evolution of phishing through shortened URLs. In their study they explained that due to the constraints of the limited text size in Twitter, attackers use URL shortening services to hide their identity. So, they designed a phishing detection system on Twitter, comprising three steps. The first is to extract from PhishTank database, the phishing URLs detected on 2010. Then, in the second step, they linked these URLs to bit.ly services to have the corresponding shortened phishing URLs. In the last step, they analyzed the tweets containing the relative URLs, using several analysis methods (analysis of user-profiles, analysis of the text of the Tweet, analysis of friends and followers).

In their work [7], the authors developed a real-time phishing detection system on Twitter called PhishAri. For this, they exploited features specific to Twitter, such as the content of the tweet and its characteristics such as the length of the text, hashtags, and mentions as well as the functionality of URLs. Similarly, the authors used the features of the user posting the tweet, such as account age, a number of tweets, and followers-followee ratio. Regarding the classification techniques deployed to detect phishing tweets, the researchers used machine learning methods specifically: naive Bayesian classifier, decision tree, and decision tree forests. For the deployment of their system they have developed an extension on the Chrome browser, which classifies in real-time a tweet as being either a phishing attempt or not.

Moreover, [9] proposed a Web framework for the detection of phishing tweets in real-time. Thus, they developed a system based on machine learning, using three types of features to improve detection accuracy. The first is based on the properties of URLs such as the length of the URL, the number of points it contains, the number of subdomains, and other properties. The second relies on the features of tweets such as the length of the tweet, the number of @tags, the position of the #tags, and the number of RTs (retweet). As for the third, it is based on Whois to identify the owner of the domain name and the date of creation.

Also, in their research work [10], the authors implemented a distributed architecture to solve the problem of phishing on Twitter. To do this, they started by collecting tweets via the Twitter API, extracting URLs from the tweets, then combining 12 features of the URL and several features linked to the tweet, to train their classifier algorithm. To implement this work, they created a browser extension allowing the detection of phishing for Twitter users. So, the browser extension protects the user from phishing attacks by adding a red flag to phishing tweets.

Likewise, the authors in [14] carried out a study aimed at improving classification features using machine learning, to classify a set of data collected from Twitter. In this study, three supervised machine learning techniques were used: SVM, KNN (K-Nearest Neighbours), and Random Forest. The result of this study showed that with only eleven features selected, a classification accuracy of 94.75% was obtained which is higher than that obtained by other researchers who had used more than eleven features for the same set of data collected on Twitter i.e "94.56%". They also found that RF is the best classification model compared to SVM and KNN.

In continuation of their work cited in [14], the authors proposed in [17] a security alert mechanism to report tweets containing phishing URLs in real-time, for Twitter users. To do this, they used the eleven classification features identified above and their RF classification model, to implement their alert mechanism. So, they managed to send security alerts to twitters in real-time.

Furthermore, in their work [15], the authors presented an approach based on social engineering using natural language processing techniques (NLP), to analyze the text and detect inappropriate messages, which are indicators of phishing attacks. Their approach consists of carrying out a semantic analysis of the text transmitted by the attacker to verify whether a sentence interrogates sensitive information or prompts for the execution of action likely to divulge personal information. For learning, they used phishing and non-phishing emails. They achieved results with a 95% accuracy.

Similarly, in [16] the researchers designed several features based on the domain name and trained a machine learning model using a dataset from the OpenPhish.com platform, to test suspicious websites. The learning model has seven features and has achieved a classification accuracy of 94%.

Moreover, in [12], the researchers developed a real-time phishing detection system on Twitter, called PDT (Phishing Detector for Twitter) based on unsupervised learning. This two-phase system combines a clustering algorithm (unsupervised learning) with a DerTIA algorithm used to detect attacks on the twitter platform. The technique used in PDT exploits features specific to Twitter as well as URLs features. They too relied on features specific to the account of the user posting the tweet.

Finally, in [11] the authors developed a real-time spam detection system for Twitter called "A Large Ground Truth for Timely Twitter Spam Detection", based on machine learning. For this, they collected more than 600 million public tweets. Then they tagged around 6.5 million spam tweets and extracted some features that can detect spam in real-time. In addition, they offered features based on information from the user posting the tweet, as well as other features based on messages (Tweets). In this work, they obtained an F-measure of 93.6% with the Random Forest classification algorithm.

In Table 1 we have summarized the works presented above. We find that most of them have used machine learning techniques based on analysis of the URL, the content of the Tweet, or the combination of both, with the consideration of the user account for some. We note that one work is based on the blacklist of shortened phishing URLs (PhishTank) [4].

By studying the literature, our attention is caught to the fact that most of previous works cited above are only based on machine learning methods, without adding the blacklist approach, which add another filtering layer and could eliminate a large number of false positives (phishing URLs). Besides that, these contributions have not addressed the shortened URLs issue, which means that if someone shortens a phishing URL, it will neither be detected by their systems nor by the Twitter detection system. Likewise, most of the cited research did not analyze the user having posted the tweet. Finally, the features of phishing and the techniques of this attack are constantly evolving, which questions the relevance and the effectiveness of the features over the years. Hence the need to carry out a study and tests to bring out the most relevant features currently.

This led us to propose a new approach, where we combined the method based on using phishing URLs blacklists and machine learning techniques as well as taking into consideration the user account.

TABLE I. SUMMURY OF THE VARIOUS RESEARCH WORKS ON PHISHING

Authors	Method of detection	Feature types	Precision
Chhabra et al. (2011) [4]	Blacklist of shortened phishing URLs (phishtank)	/	/
Aggarwal et al. (2012) [7]	Supervised learning (Naive bayes, Decision tree, RF)	Based: - URL: 6 - Whois: 3 - Tweet: 6 - Twitter account:7	Naïve Bayes: 87.02% Decision tree: 89.28% RF: 92.52%
Sharma et al. (2014) [9]	Supervised learning	Based: - URL: 7 - Tweet: 5 - Whois : 2	94.56%
Nair et Prema (2014) [10]	Supervised learning	Based : - URL : 12 - Tweet - Information : 5	92,20%
Liew et al. (2018, 2019) [14], [17]	Supervised learning	Based: - URL: 11	94.75%
Peng et al. (2018) [15]	Semantic text analysis (Multinomial Naive Bayes)	/	95%
Shirazi et al. (2018) [16]	Supervised learning	Based: - Domain name - URL : 7	97.7 %
Jeong et al. (2016) [11]	Supervised learning (spam detection)	Based: - Twitter account: 6 - Tweet: 6	90,71 %.
Chen et al. (2015) [12]	Unsupervised learning	Based: - Twitter account: 6 - Tweet: 8 - URL: 5	93,63%

III. Our proposal

Our approach for analyzing and detecting phishing content on twitter is based on three steps (see figure 1), in the first one we verify the occurrence of the URLs, contained in a tweet, in a database of malicious URLs (Blacklisting) [4]. Then, if it is not the case, we proceed to the second step which consists in using supervised machine learning methods to classify the URL (URL analysis using machine learning), and therefore the tweet, in one of the categories (phishing or legitimate) according to a set of features and a leveraging prediction model applied to URLs. In the third step, we proposed a user analysis mechanism to lessen the spread of phishing links on Twitter (Twitter User account analysis).

By comparing our approach with the work presented in chapter II, our contribution can be summarized in the following points:

(1) We have added a step consisting in checking the existence of the tested URL in PhishTank blacklist, which is new compared to the ones used in cited contributions, this eliminates existing URLs in the blacklist database.

(2) Our analysis system checks if the URL contained in the tweet has been shortened and if so, retrieves the original URL for analysis.

(3) After an in-depth study and several tests with taking into consideration the evolution of phishing attack techniques, we have chosen twenty-five features to identify phishing URLs. Some of these features have not been used in other contributions such as IP address, URL shortening services, presence of number port, Iframe redirection, Spy form, Server form manager (SFH), DNS (Domain Name System) registration.

(4) We opted for six (6) features for the analysis of Twitter accounts. These six features concern the user accounts and are completely different from the features of phishing URLs.

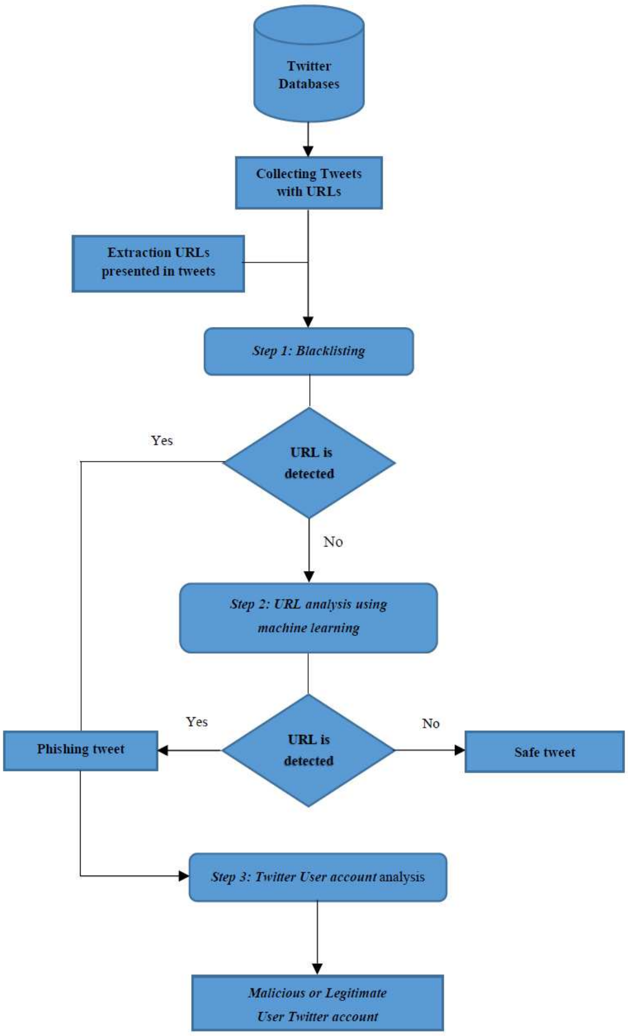

Twitter Databases
Collecting Tweets with URLs
Extraction URLs presented in tweets
Step 1: Blacklisting
Yes
URL is detected
No
Step 2: URL analysis using machine learning
Yes
No
Phishing tweet
URL is detected
Safe tweet
Step 3: Twitter User account analysis
Malicious or Legitimate User Twitter account

Fig. 1. Global schema of the proposed approach

A. Step 1: Verification in the PhishTank database

In the first step of our approach, which consists of verifying the tweet's URL in PhishTank database [18], we start by extracting the URLs from the text of the tweet, then we interact with PhishTank database to proceed with the verification of these URLs. If the URL is detected as a phishing URL, an alert signal is displayed, otherwise, we go to the second step detailed next.

This phase aims to curtail the false negative rate, therefor it is considered as a preprocessing step for the machine learning process.

B. Step 2: Classification based on URLs analysis using machine learning

The links verification in a database cannot detect new phishing sites that have not yet been added to that database. According to our research, Twitter uses Google's secure browsing to detect and block phishing links [12]. Although this verification can block malicious URLs, its effectiveness is limited, since verification services take an average of four (04) days to add a new link to the database, while most tweet accesses take place in both days after publication [11], which prevents users from being protected in real-time. This was the reason that motivated us to complete our approach by the classification based on URLs analysis using machine learning.

Our URL classification process is subdivided into two stages A and B. The stage A is carried out only once during the learning process, while the stage B will be applied, in the experimentation part, whenever a tweet is collected.

In step A, we used a set of selected data containing malicious and legitimate URLs. Then we chose the features, some of which were not used in previous work. Then we extracted these features and represented them in a data table. Next, we cleaned up this data and finally built the classification model. Step B concerns the extraction of the URLs presented in tweets from the tweets collecting (with URLs), verifying if it has been shortened, retrieve the original URL if so, and then represent it according to the classification model defined in step A.

1) Learning data: After several searches, we could not find a reliable dataset consisting of a set of phishing tweets because the data used in these searches are not rendered public. We also thought about creating our dataset from phishing tweets, but this task is very arduous. So, to implement our classification model, we used the UCI Machine Learning phishing dataset.

2) Features selection: There are hundreds of features related to phishing websites in the literature. However, these features are not all important and effective, therefore, using the maximum of these features does not give us a better classification accuracy.

Therefore, after having studied various phishing features, we started our experiments with thirty features, which we considered the most relevant with phishing websites. Nevertheless, we have found that the selected features are not all of the same degree of importance and some have become obsolete over time and the evolution of techniques used by phishers.

Indeed, to choose the best features we carried out several iterations of learning with several combinations of features, including the new features that have been introduced in this article, and each time we observed the results obtained to evaluate the importance of each characteristic and kept at the end, the combination that gave us the best accuracy rate. Figure 2 display the occurrence frequency (horizontal axis) of each features (vertical axis) in our dataset. This shows us the impact of each feature on the learning process. These features are divided into four categories (see table 2).

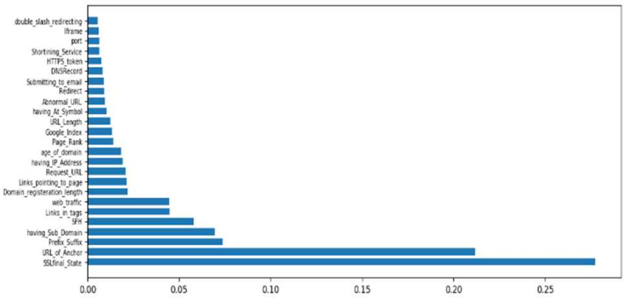


Fig. 2. Occurrence frequency of selected features

3) Features extraction and representation of URLs: The feature extraction is done once the URL has been retrieved from the tweet. The set of features is represented by a vector of integers (comprising 1, -1 and 0), each box of which contains a feature represented by the following function CI:

$$\text{CI} \begin{cases} = 1 \text{ Presence of feature in the URL} \\ = 0 \text{ Suspect} \\ = -1 \text{ Absence of feature in the URL} \end{cases}$$

TABLE II. DIVISION OF FEATURES INTO FOUR CATEGORIES

Category 1: Address bar features	**Category 2: Features based on HTML and JavaScript**	**Category 3: Domain-based features**	**Category 4: Features dissimilar on recognized rules of a URL**
Using the IP Address in place of the domain name in the URL	Website Forwarding	Age of domaine	Spy form
Long URL to Hide the Suspicious Part	IFrame Redirection	Website ranking	Website identity
URL's having "@" Symbol		PageRank	Server Form Handler (SFH)
The existence of HTTPS		DNS record	URL of Anchor
Redirecting using "//"		Google Index	Using Pop-up Window
Adding Prefix or Suffix Separated by (-) to the Domain		Number of Links Pointing to Page	Statistical-Reports Based Feature
Domain Registration Length			
The Existence of "HTTPS" Token in the Domain Part of the URL			
Using Non-Standard Port			
Using URL Shortening Services "TinyURL"			
Sub Domain and Multi Sub Domains			

The algorithm 1 shows the extraction and representation of a feature from an URL.

The function rule_i(URL) checks if the concerned feature (i) is present in the URL or not. Every feature has a specific function to verify its presence.

Algorithm 1 Feature extraction and representation.

```
Input: URL;
Output: A feature i;
Feature_i (url) function;
Begin
  if (rule_i (url) = true) then
    return (1);
  else
    if (rule2_i (url) = true) then
      return (0);
    else
      return (-1);
    end if;
  end if;
End;
```

Regarding the extraction of the twenty-five features for the construction of the integer vector (representation of the URL), it is presented by the pseudo algorithm 2.

Algorithm 2 Extraction of all the features for representation of the URL.

```
Input: URL;
Output: Feature vector;
Url_extraction function (url);
Begin
  int Tab [N] = [];
  for i=1 to N do
    Tab[i] = feature_i(url);
  end for;
  Return (Tab);
End;
```

4) Data cleaning: The purpose of this phase is to resolve anomalies and reduce the size of the dataset and therefore the learning time. In our case we found that there is in the data URLs having the same features. Therefore, we eliminated all redundant lines as well as the columns of useless attributes.

5) Classification models: In our case, we used three classification algorithms to generate the classification model, Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM). The generated model is stored for use in our phishing tweet prediction system.

The main parameters that gave us the best result in each of the classifiers are as follows:

LR: Random state = 0.

RF: Criterion = Gini impurity , The number of features to consider when looking for the best split = log2(nb_fatures), The number of trees in the forest = 270, Random state = 0.

SVM: Regularization parameter = 1, kernel type = RBF, Kernel coefficient = 0.2, Random state = 0.

C. Step 3: Analysis of Twitter User accounts

The third step of our approach is the analysis of user accounts. For this, we used a machine learning method via a set of data collected from the Twitter social network with features related to the users of this social network. The user classification process goes through the same steps of the previous approach (classification of phishing links).

1) Learning data: Among the few datasets containing information on Twitter accounts, we used a dataset in [11] which contains 10,000 user accounts collected from Twitter.

2) Features selection: There are several pieces of information that can be collected from the Twitter API on a user account. This information can be used as features for the learning, however in order to achieve satisfactory classification accuracy, we must attentively choose all of the features.

Therefore, we started our experiments with numerous features, and as the URL classification process, we used several combinations of these features to bring out the most relevant features; at the end we opted for six (6) features, which gave the best results. These are account age, number of followers, number of subscriptions, number of favorites, number of times the user has been listed, and number of tweets. These features are extracted from the "user" object contained in the JSON object fetched via the Twitter API. A vector of integers will be used to represent all these features.

3) Classification models: To build a classification model we used the Random Forest classification algorithm. Our malicious user prediction model leverages this generated model to conduct user classification.

IV. Implementation and Evaluation

In this section, we will present the learning and test data that we used to assess the three stages of our approach. Next, we will describe the validation method and the measures considered to assess the prediction rate. Then, we will discuss and interpret the results obtained and finally, show the tests and demonstration of the application.

A. Partition of learning and test data

The dataset we used for the detection of phishing attacks is the "UCI phishing dataset", built from three different sources: PhishTank [18], MillerSmiles [19] and Google. It contains 11054 instances including 6157 phishing cases and 4898 legitimate (safe). We note that the data of phishtank database used in step 1 is not the same one contained in this dataset.

The datasets we used to classify twitters as "malicious" or "legitimate" contain ten thousand user accounts [11], among which five thousand are classified as "malicious" and five thousand as "non-malicious". We have divided theses datasets into two parts. One part contains 75% for learning and the other part contains the remaining 25% for the test.

B. *Validation method and evaluation criteria*

To evaluate the performance of the generated model for the detection of phishing URLs, we used the cross-validation method. On the other hand, we applied the confusion matrix for assessing the effectiveness of the classification.

In our case we have a binary classifier that predicts two classes denoted class -1 (legitimate URL) and class 1 (phishing URL).

Regarding quality measures, we have chosen the most used measures in the literature to evaluate a classifier: recall, precision, and F-measure. Since in our case, we have two classes (phishing and non-phishing), the overall averages of precision and recall over the two classes can be appraised by the macro-average.

C. *Experiments and results*

We present, in this section, the prediction results obtained, with the different classifiers, for phishing URLs (Table 3, 4, and 5), then for malicious users.

1) Prediction results for malicious users: Concerning the prediction results of malicious users, after several tests, we managed to reach the best accuracy, which is 74.96% with the Random Forest classifier.

D. *Comparison and interpretation of results*

As shown in Figure 3, the prediction results of phishing URLs with the Logistic Regression method were acceptable with an accuracy of 90.28%, but the results obtained by the Support Vector Machine method are more interesting with an accuracy of 93.43%. From these results also, we see that the Random Forest method gives the apical precision, which is 95.51%.

This is explained by the fact that the Logistic Regression method does not have the capacity to deal with large dimensionality and therefore, has not given better accuracy. Regarding Support Vector Machine, it gave acceptable results but not the best, since it cannot process large datasets, like the one we used, with very high observation. The use of Random Forest slightly improved the results, because it gives a good performance in prediction since it combines regression and classification, which saves time on data preparation.

TABLE III. PREDICTION RESULTS FOR PHISHING URLS USING LOGISTIC REGRESSION.

Iteration	Class	Precision	Recall	F Measure
1	Phishing	0.80	0.90	0.85
	Safe	0.93	0.86	0.90
2	Phishing	0.89	0.88	0.89
	Safe	0.93	0.93	0.93
3	Phishing	0.90	0.90	0.90
	Safe	0.94	0.94	0.94
4	Phishing	0.84	0.91	0.88
	Safe	0.94	0.89	0.92
5	Phishing	0.90	0.83	0.87
	Safe	0.90	0.94	0.92
Average	Phishing	86.63	88.75	87.57
	Safe	92.80	91.20	92.20
Global Precision: 90.28%				

TABLE IV. PREDICTION RESULTS FOR PHISHING URLS USING SVM

Iteration	Class	Precision	Recall	F Measure
1	Phishing	0.91	0.90	0.91
	Safe	0.94	0.94	0.94
2	Phishing	0.94	0.90	0.92
	Safe	0.94	0.96	0.95
3	Phishing	0.94	0.93	0.94
	Safe	0.96	0.96	0.96
4	Phishing	0.88	0.94	0.91
	Safe	0.96	0.92	0.94
5	Phishing	0.93	0.87	0.90
	Safe	0.92	0.96	0.94
Average	Phishing	92.00	90.94	91.42
	Safe	94.40	94.80	94.60
Global Precision: 93.43%				

TABLE V. PREDICTION RESULTS FOR PHISHING URLS USING RANDOM FOREST

Iteration	Class	Precision	Recall	F Measure
1	Phishing	0.95	0.93	0.94
	Safe	0.96	0.97	0.97
2	Phishing	0.95	0.93	0.94
	Safe	0.95	0.97	0.96
3	Phishing	0.94	0.93	0.93
	Safe	0.95	0.96	0.95
4	Phishing	0.93	0.95	0.94
	Safe	0.97	0.96	0.96
5	Phishing	0.97	0.95	0.96
	Safe	0.97	0.98	0.98
Average	Phishing	94.70	93.63	94.15
	Safe	96.00	96.80	96.40
Global Precision: 95.51%				

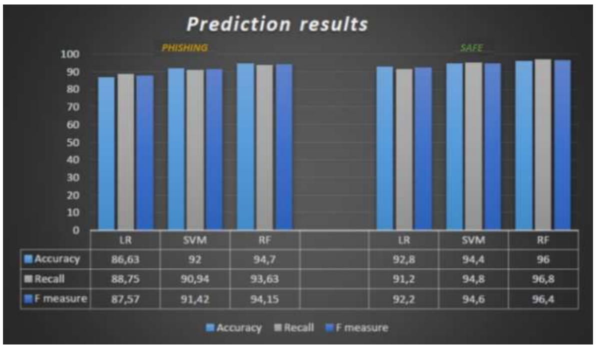


Fig. 3. Comparison of results

The average accuracy of each classifier is presented in the following figure 4:

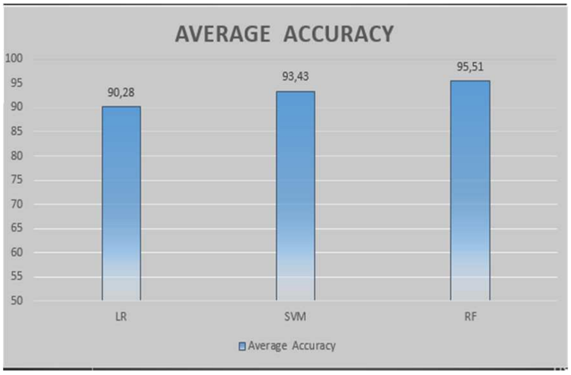


Fig. 4. Average accuracy

If we compare our work with the other existing works, notably those of [7], [9], [12] and [17], although we used different datasets from each other, we obtained a better accuracy which reached 95.51% with the Random Forest classifier, well above 92.52% obtained in [7] and 93.63% reached in [12], and little more than that of [9] which is 94.56% and the 94.75% of [17].

E. *Tests and demonstration of the application*

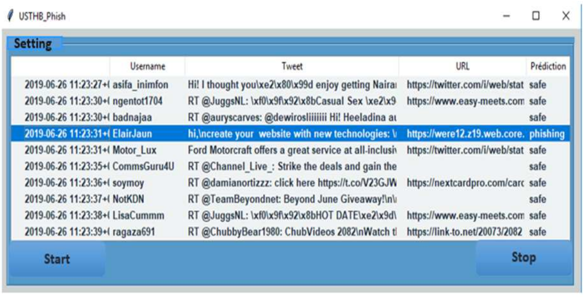
We have implemented our approach in a tool that help researches and end users to detect phishing tweets. Indeed, the application allow us to display an alert (on the screen) to warn that there is a phishing tweet posted on twitter. In addition, it displays account features of tweets.

At this stage, the classification model has been integrated into our tool to detect tweets containing a phishing URL in real-time. Also, to test our application, we extracted a hundred phishing URLs from the "isitphishing.com" [20] and "PhishTank" websites. We have also created three Twitter accounts to post these URLs. Then, we posted these URLs on Twitter one by one, where the result of each prediction for the phishing URL was observed and recorded accordingly. This experiment aims to test the tool and show the efficiency of our approach by combining the 3 steps explained previously. The results obtained are explained in table 6.

TABLE VI. APPLICATION TEST RESULTS.

Description	**Number of phishing URLs**	**Number of URLs detected correctly**	**Number of URLs not detected**
Collected from phishtank.com	50	50	0
Collected from isitphishing.org	50	45	5
Total	100	95	5

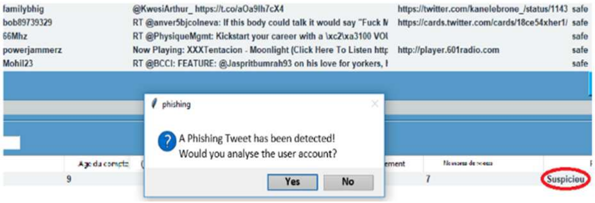
An interface of our developed application, allowing the detection of a phishing tweet is shown in the figure 5.



	Username	Tweet	URL	Prédiction
2019-06-26 11:23:27+(	asifa_inimfon	Hi! I thought you\xe2\x80\x99d enjoy getting Nairai	https://twitter.com/i/web/stat	safe
2019-06-26 11:23:30+(	ngentot1704	RT @JuggsNL: \xf0\x9f\x92\x8bCasual Sex \xe2\x9	https://www.easy-meets.com	safe
2019-06-26 11:23:30+(	badnajaa	RT @auryscarves: @dewiroslііііііі Hi! Heeladina au		safe
2019-06-26 11:23:31+(	ElairJaun	hi,\ncreate your website with new technologies: \	https://were12.z19.web.core.	phishing
2019-06-26 11:23:31+(	Motor_Lux	Ford Motorcraft offers a great service at all-inclusiv	https://twitter.com/i/web/stat	safe
2019-06-26 11:23:35+(	CommsGuru4U	RT @Channel_Live_: Strike the deals and gain the		safe
2019-06-26 11:23:36+(	soymoy	RT @damianortizzz: click here https://t.co/V23GJW	https://nextcardpro.com/carc	safe
2019-06-26 11:23:37+(	NotKDN	RT @TeamBeyondnet: Beyond June Giveaway!\n\		safe
2019-06-26 11:23:38+(	LisaCummm	RT @JuggsNL: \xf0\x9f\x92\x8bHOT DATE\xe2\x9d\	https://www.easy-meets.com	safe
2019-06-26 11:23:39+(	ragaza691	RT @ChubbyBear1980: ChubVideos 2082\nWatch tl	https://link-to.net/20073/2082	safe

Fig. 5. Detection of a phishing tweet

So, if a phishing tweet is detected, an alert is raised to warn the user that there is a phishing tweet that has been posted (see Figure 6).



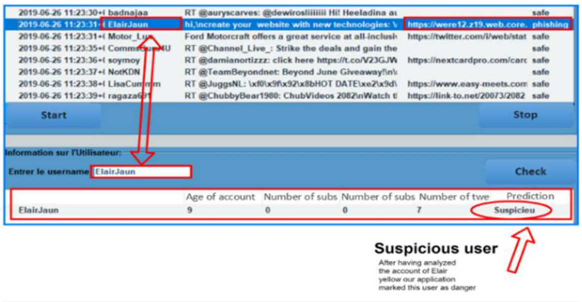
familybhig	@KwesiArthur_ https://t.co/aOa9lh7cX4	https://twitter.com/kanelebrone_/status/1143	safe
bob89739329	RT @anver5bjcolneva: If this body could talk it would say "Fuck N	https://cards.twitter.com/cards/18ce54xher1/	safe
66Mhz	RT @PhysiqueMgmt: Kickstart your career with a \xc2\xa3100 VOI		safe
powerjammerz	Now Playing: XXXTentacion - Moonlight (Click Here To Listen http	http://player.601radio.com	safe
Mohil23	RT @BCCI: FEATURE: @Jaspritbumrah93 on his love for yorkers, I		safe

A Phishing Tweet has been detected!
Would you analyse the user account?

Age du compte	ement	Nombre de [illegible]	
9	7		Suspicieu

Fig. 6. Phishing alert

The last step of our application is the analysis of the user account (see Figure 7).



2019-06-26 11:23:30+(	badnajaa	RT @auryscarves: @dewiroslііііііі Hi! Heeladina au		safe
2019-06-26 11:23:31+(	ElairJaun	hi,\ncreate your website with new technologies: \	https://were12.z19.web.core.	phishing
2019-06-26 11:23:31+(	Motor_Lux	Ford Motorcraft offers a great service at all-inclusi	https://twitter.com/i/web/stat	safe
2019-06-26 11:23:35+(	CommsGuru4U	RT @Channel_Live_: Strike the deals and gain the		safe
2019-06-26 11:23:36+(	soymoy	RT @damianortizzz: click here https://t.co/V23GJW	https://nextcardpro.com/carc	safe
2019-06-26 11:23:37+(	NotKDN	RT @TeamBeyondnet: Beyond June Giveaway!\n\		safe
2019-06-26 11:23:38+(	LisaCummm	RT @JuggsNL: \xf0\x9f\x92\x8bHOT DATE\xe2\x9d\	https://www.easy-meets.com	safe
2019-06-26 11:23:39+(	ragaza691	RT @ChubbyBear1980: ChubVideos 2082\nWatch tl	https://link-to.net/20073/2082	safe

	Age of account	Number of subs	Number of subs	Number of twe	Prediction
ElairJaun	9	0	0	7	Suspicieu

Suspicious user
After having analyzed
the account of Elair
yellow our application
marked this user as danger

Fig. 7. Prediction of a suspicious user

If the account is classified as suspicious then the user is probably malicious. After this phase, we intend to add a functionality that report this user to Twitter.

V. CONCLUSION AND PERSPECTIVES

In this work, we have tackled the problem of phishing attacks on social networks. First, we described the different methods and techniques used in this attack. Next, we discussed the various research work concerning phishing detection. Then, we proposed an approach comprising of three steps, which are the verification in a blacklist, the analysis of URLs, and the analysis of user accounts. To do this, we implemented three classification models for URL analysis by integrating new features compared to existing work, and then compared the results obtained by each classifier. We have obtained an accuracy that exceeds 95% with the Random Forest classifier. For the analysis of user accounts, we obtained an accuracy of around 75% using the Random Forest classifier. Finally, we implemented our approach in the form of an application.

As research perspectives, we plan to: (1) Add other parameters to the characteristic vectors, such as the features linked to social engineering techniques (analysis of the content of the tweet); (2) Apply this approach to other larger datasets, to ensure scalability; (3) Integrate the application produced as a module in a real-time detection system with the aim of analyzing tweets with different combinations possible and (4) Generalize this approach to other social networks, with its application to other attacks.

REFERENCES

[1] Ma, J., Saul, L. K., Savage, S., & Voelker, G. M.: Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1245-1254 (2009).

[2] Ma, J., Saul, L. K., Savage, S., & Voelker, G. M.: Identifying suspicious URLs: an application of large-scale online learning. In Proceedings of the 26th annual international conference on machine learning, pp. 681-688 (2009).

[3] Sanglerdsinlapachai, N., Rungsawang, A.: Using domain top-page similarity feature in machine learning-based web phishing detection. In 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 187-190. IEEE (2010, January).

[4] Chhabra, S., Aggarwal, A., Benevenuto, F., Kumaraguru, P.: Phi. sh/$ ocial: the phishing landscape through short urls. In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, pp. 92-101 (2011).

[5] Basnet, R. B., Sung, A. H., Liu, Q.: Rule-based phishing attack detection. In Proceedings of the International Conference on Security and Management (SAM) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2011).

[6] Pandey, M., Ravi, V.: Detecting phishing e-mails using Text and Data mining". IEEE International Conference on Computational Intelligence and Computing Research (2012).

[7] Aggarwal, A., Rajadesingan, A., Kumaraguru, P.: PhishAri: Automatic realtime phishing detection on twitter. In 2012 eCrime Researchers Summit, pp. 1-12. IEEE (2012).

[8] Moore, T., Clayton, R.: Discovering phishing dropboxes using email metadata. In 2012 eCrime Researchers Summit, pp. 1-9. IEEE (2012).

[9] Sharma, N., Sharma, N., Tiwari, V., Chahar, S., Maheshwari, S.: Real-Time Detection of Phishing Tweets. In Fourth Int Conf Comput Sci Eng Appl, pp. 215-27 (2014).

[10] Nair, M. C., Prema, S.: A Distributed System for Detecting Phishing in Twitter Stream. Int J Eng Sci Innov Technol, 3(2), pp. 151-158 (2014).

[11] Chen, C., Zhang, J., Chen, X., Xiang, Y., & Zhou, W.: 6 million spam tweets: A large ground truth for timely Twitter spam detection. In 2015 IEEE international conference on communications (ICC), pp. 7065-7070 (2015).

[12] Jeong, S. Y., Koh, Y. S., Dobbie, G.: Phishing detection on Twitter streams. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 141-153). Springer, Cham (2016).

[13] Paolo, G.: Tweets and the streets: Social media and contemporary activism. Pluto Press, (2018).

[14] Liew, S.W., Sani, N. F. M., Abdullah, M. T., Yaakob, R., Sharum, M. Y.: Improvement of classification features to increase phishing tweets detection accuracy. Journal of Theoretical & Applied Information Technology, 96(10) (2018).

[15] Peng, T., Harris, I., & Sawa, Y.: Detecting phishing attacks using natural language processing and machine learning. In 2018 IEEE 12th international conference on semantic computing (icsc), pp. 300-301 (2018).

[16] Shirazi, H., Bezawada, B., & Ray, I.: "Kn0w Thy Doma1n Name" Unbiased Phishing Detection Using Domain Name Based Features. In Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, pp. 69-75 (2018).

[17] Liew, S. W., Sani, N. F. M., Abdullah, M. T., Yaakob, R., Sharum, M. Y.: An effective security alert mechanism for real-time phishing tweet detection on Twitter. Computers & Security, 83, 201-207 (2019).

[18] https://www.phishtank.com/faq.php consulté le 10/09/2019.

[19] http://www.millersmiles.co.uk/ consulté le 11/11/2019.

[20] https://isitphishing.org/ consulté le : 15/12/2019.

[21] https://securelist.com/spam-and-phishing-in-q1-2020/97091/ consulté le 28/04/2020.