Human-centric Computing and Information Sciences

RESEARCH Open Access

A novel lightweight URL phishing detection system using SVM and similarity index

Mouad Zouina[1*] and Benaceur Outtaj[2]

*Correspondence: zouina.mouad@gmail.com
[1] TSE Research Team, ENSIAS, Mohammed V University of Rabat, Rabat, Morocco
Full list of author information is available at the end of the article

Abstract

The phishing is a technique used by cyber-criminals to impersonate legitimate websites in order to obtain personal information. This paper presents a novel lightweight phishing detection approach completely based on the URL (uniform resource locator). The mentioned system produces a very satisfying recognition rate which is 95.80%. This system, is an SVM (support vector machine) tested on a 2000 records data-set consisting of 1000 legitimate and 1000 phishing URLs records. In the literature, several works tackled the phishing attack. However those systems are not optimal to smartphones and other embed devices because of their complex computing and their high battery usage. The proposed system uses only six URL features to perform the recognition. The mentioned features are the URL size, the number of hyphens, the number of dots, the number of numeric characters plus a discrete variable that correspond to the presence of an IP address in the URL and finally the similarity index. Proven by the results of this study the similarity index, the feature we introduce for the first time as input to the phishing detection systems improves the overall recognition rate by 21.8%.

Keywords: Phishing, Phishing detection system, Web security, SVM, URL, Hamming distance

Background

The phishing is a technique used by cybercriminals to mimic legitimate websites in order to obtain personal information such as login, password and credit card number which leads to an identity theft. However these criminals typically use phishing to subtract money; for that purpose they target online banking, online payment systems, e-commerce (electronic commerce) websites and m-commerce (mobile commerce) applications.

Despite all efforts made to counter the phishing threat, this attack still manage to cause serious damage, according to the FBI (Federal Bureau of Investigation) [1] the phishing attack cost $1.2 billion in the span of a year and 2 months between 1st October 2013 and 1st December 2014. Furthermore the colossal financial losses aren't the only damages caused by the phishing attack since the number of phishing websites detected by the anti-phishing working group [2] increased by 250% from the last quarter of 2015 to the first quarter of 2016 moreover the number of unique phishing websites detected between January and March 2016 is 289,371 which is more than enough of a reason to make us question whether the current anti-phishing systems are efficient?

Detecting the phishing attack proves to be a challenging task. This attack may take a sophisticated form and fool even the savviest users: such as substituting a few characters of the URL with alike unicode characters. By cons, it can come in sloppy forms, as the use of an IP address instead of the domain name.

Nonetheless, in the literature, several works tackled the phishing attack detection challenge while using artificial intelligence and data mining techniques [5–9] achieving some satisfying recognition rate peaking at 99.62%. However those systems are not optimal to smartphones and other embed devices because of their complex computing and their high battery usage, since they require as entry complete HTML pages or at least HTML links, tags and webpage JavaScript elements some of those systems uses image processing to achieve the recognition. Opposite to our recognition system since it is a less greedy in terms of CPU and memory unlike other proposed systems as it needs only six features completely extracted from the URL as input.

In this paper, after a summary of this field key researches, we will detail the characteristics of the URL that our system uses to do the recognition. Otherwise we will describe our recognition system, next in the practical part we will test the proposed system while presenting the results obtained. Last but not least we will enumerate the implications and advantages that our system brings as a solution to the phishing attack.

Related works

In the literature the cyber attack called phishing is treated in three different ways.

One of the approaches to counter phishing is the blacklist, that blacklist contains known phishing websites acquired by techniques such as user votes, those blacklists are typically deployed as plug-ins in browsers in order to check each URL entry in the blacklist. Then it prevents the user whenever he attempts a connection to one of these malicious websites which are included in the blacklist. To cite some examples: internet explorer phishing filter [3], google safe browsing for Firefox [4]. However this approach still facing an issue since it offers no protection against the new phishing websites that are not included in the blacklist. Not to mention the slow update process of the blacklist and the typical short duration (some hours) of the phishing websites.

Other researchers have opted for the use of artificial intelligence and data mining to detect the phishing websites. This is the path that is most exploited and gives far more promising results and wherein our work falls. The development of intelligent systems for the detection of phishing websites have been the subject of many researches like CANTINA+, the work of Xiang et al. [5] which is a phishing websites detection platform based on the URL characteristics and query results through search engines in addition to some elements of HTML (hypertext markup language) pages. While on subject CANTINA+ obtained a recognition rate of 92%. Moreover Fu et al. [6] have proposed a detection system based on the visual similarity of web pages calculated by earth mover's distance. Other researchers use imaging techniques to detect phishing as Li et al. [7] hybrid system that used the image detection system PSO-SVM (particle swarm optimization support vector machine) to achieve a recognition rate of 99%. This system sends the same query to two different DNS (domain name system) server to compare their returned results. But despite the impressive recognition rate of this technique an attacker can tamper with the results of the two DNS servers using a man in the middle attack and therefore corrupt the recognition of all the system. Thomas et al. [8] developed a real-time spam and phishing detection system, their system uses several criteria such as the characteristics of the URL, the number of redirects, web pages HTML elements and JavaScript, geo-location data, and DNS data. To perform the phishing website detection their technique needs web pages HTML elements and JavaScript, a task that will be impossible if the attacker blocks the IP (internet protocol) of their Crawler from collecting the needed data. As well as the work of Jeeva et al. [9] This phishing detection system acts within two phases, the first procedure leads to a research of the suspect URL in the white list called repository once this last is present in the list, the URL is deemed legitimate, however if the URL doesn't exist in the repository then its subjected to further examination during the second phase of the recognition which consists of an association rule mining algorithm. Finally the research of Ramesh et al. [10] which reached an impressive recognition rate of 99.62%. This mentioned system uses a suspicious web page keywords as an input to a search engine to get links, and then compare them with the links within the suspicious web page to keep only the existing links as an input to TID algorithm (target identification algorithm) and finally a DNS lookup is performed to check the domain name of the targeted website with its IP address and there lies the weak link of this proposal and make it vulnerable to the man in the middle attack.

The other solutions proposed to the phishing issue in the literature are works that do not try to distinguish between the legitimate and phishing websites. Oppositely they opt to consolidate user authentication in order to overcome this problem. Among other things the study by Huang et al. [11] which proposes to replace the use of a permanent password by a (one-time password) that should be provided to the user by a third party under a message form. The problem with this technique is its total dependence on the third party. In other words when the latter is under attack, the security of the entire system is compromised. Another proposal by Yue et al. [12] is to send a group of purposely wrong logins and passwords instead of the actual user login and password when connecting to a phishing website, the detection of the attack is done by a plug-in in the user's browser and this the Achilles heel of this proposal because it relies on a detection system made by a third party.

The URL based phishing detection system

Feature extraction and analysis

In the system we propose, we initially minutely observed and studied a 2000 records database including 1000 phishing website records built from the PhishTank database [13]. In this paper the targeted websites of the phishing attack are vital therefore all the retained 1000 phishing website records must contain their respective target. Moreover the studied database consist also of 1000 legitimate websites which we collected ourselves by combining Alexa's [14] 500 top global website with 500 websites resulted from queries to google search engine, as for the queries we used to feed our database are (*.bank.*, *.commerce.*, *.trade.*) in consideration of the phishing attack and the websites more likely to be targeted. Our analysis shows that the URL portion of interest is composed of several parts as shown in Fig. 1.

For the remainder of this document the word URL will designate only part 6 of Fig. 1 that is to say, we are only interested in the second-level domain name (4 in Fig. 1) and the first level domain name (5 in Fig. 1) as well as to all sub-domains except the default sub-domain (www). The first segment was removed from the interest zone, because the HTTPS certificates are not part of the scope of this work.

Figure 2 shows just the URL part that interests us.

After a preliminary study on our database, we have discarded the at (@) and the underscore (_) from the URL characteristics used in order to perform the recognition because on the totality of the dataset we found no occurrence of these two URL characteristics; thus one deduces their irrelevance. Our approach is based on artificial intelligence to detect phishing websites for this purpose we use the following URL features.

- URL_Size: this is the number of characters in the URL usually phishing websites have a more important size then legitimate websites.
- Number_of_Hyphens: this feature counts the number of the character '-' in a URL. Normally legitimate websites rarely have an occurrence of the character '-'.
- Number_of_Dots: this attribute counts the number of the character '.' (dots) in a URL (for example the number_of_dots = 4 in the following URL sub-domain2.sub-domain3.sub-domain4.mcomerce.com).
- Number_of_Numeric_Chars: we count the number of numeric characters in a URL. Since generally there is no occurrence of numeric characters in domain names of legitimate websites.
- IP_presence: this feature takes two values: 1 whenever there is an IP address in a URL otherwise 0.
- Similarity_index: the mathematically calculated distance measuring the difference between two data (two strings in our case). It is equal to 100% when measured on two identical words. Several variations and algorithms have been developed to measure this similarity among other we cite the most prevalent in this field: Levenshtein [15] Jaro Winkler [16] Normalized Levenshtein [17] longest common subsequence [18] Q Gram [19] Hamming [20].

To calculate these characteristics for each pair of phishing website and its corresponding legitimate website extracted from the database as shown in Table 1. For presentation purposes in Table 1, we coded the distance from the initial letters of their names, whether:

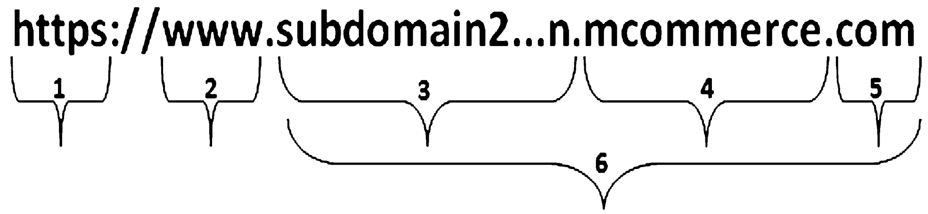

Fig. 1 URL interest zone

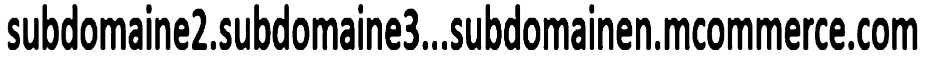

Fig. 2 URL section of interest

Table 1 Calculation of the characteristics used for the recognition of phishing websites

	URL_size	NH	ND	NNC	IP	NL	L	JW	LCS	QG	H
Min legitimate	4	0	0	0	0	0	0	1	0	0	0
Min phishing	4	0	1	0	0	0.1	1	0	2	2	1
Average legitimate	12.175	0.025	1.156	0.075	0	0	0	1	0	0	0
Average phishing	20.01	0.254	1.685	1.155	0.036	0.759	15.889	0.520	19.999	23.173	19.391
Max legitimate	31	2	3	4	0	0	0	1	0	0	0
Max phishing	202	8	14	37	1	1	192	0.959	192	192	192

- NH for Number_of_Hyphens
- ND for Number_of_Dots
- NNC for Number_of_Numeric_Chars
- IP for IP_presence
- L for the classic Levenshtein distance
- NL for Normalized Levenshtein distance
- JW for Jaro Winkler distance
- LCS for the longest common subsequence
- QG for the Q-Gram distance
- And finally H to the Hamming distance.

As shown in Table 1, the calculation of characteristics used by our system for the recognition of phishing website is done on the entire database (i.e. on 2000 records). A first reading has to infer those phishing websites:

- have an average of eight characters more than the legitimate websites,
- and may have to thirty-seven against four numeric characters only for legitimate websites.

We thought to study the relationship between these different distances metrics for visibility and comparison so we opted to calculate the correlation that may exist between them.

As shown in the last row of Table 2, the relationship between Hamming distance and Q-Gram distance, Levenshtein and longest common subsequence is manifested by a very strong correlation respectively 97, 98 and 98% when our system is tested on the entire database.

In the same course of action we thought to study the correlation that may exist between the other five URL features used in this work.

Overall as shown in Table 3, we can note the disassociation between the URL features however there is a relevant relationship between the URL_Size and the Number_of_Dots (ND) and the Number_of_Numeric_Chars (NNC) established by 75 and 63% correlation.

Phishing detection system

In this part, we will describe the characteristics used by our recognition system.

Support vector machine as well known as SVM is a supervised classification algorithm that can solve classification problems as well as regression problems. SVM was developed in 1995 [21] based on statistical learning theory by Vapnik–Chervonenkis.

Table 2 Correlation among the similarity distances in the database

	NL (%)	L (%)	JW (%)	LCS (%)	QG (%)	H (%)
NL	100	62	95	72	76	66
L	62	100	52	98	96	98
JW	95	52	100	63	68	58
LCS	72	98	63	100	91	98
QG	76	96	68	99	100	97
H	66	98	58	98	97	100

Table 3 Correlation among the other URL features in the database

	URL_Size (%)	NH (%)	ND (%)	NNC (%)	IP (%)
URL_Size	100	42	*75*	*63*	−3
NH	42	100	34	26	−3
ND	75	34	100	60	20
NNC	63	26	60	100	44
IP	−3	−3	20	44	100

Italic values indicate an important correlation between the URL_Size, ND, NNC

The kernel we used for our system is the Gaussian kernel rather known as RBF (radial basis function).

Let x and x′ two samples of the RBF kernel is defined as the following:

$$K\left(X, X'\right) = \exp\left(-\frac{\| X - X' \|^2}{2\sigma^2}\right)$$

Knowing that $X - X'^2$ is the Euclidean square distance between the two feature vectors and σ is a constant.

Moreover to validate the system we have chosen to use the fivefold cross-validation model.

In this model the database is randomly split into five equal sub-samples, from the five similar sub-samples a single sub-sample is retained for the final validation of the system while the other four sub-samples are used to train the model. Thus the cross-validation is repeated five times and each subsample is used for validation. After the final validation of the model, a single estimation is calculated which is the average estimation of the five iterations.

Figure 3 illustrate our phishing detection system.

Test of the system on the BD

In this section, we will describe the procedure of our tests and then we will present and interpret the results of these tests.

Tests

To extract the necessary characteristics to detect the phishing websites we have developed our own program to ensure the extraction of those features from the URL link and its respective target.

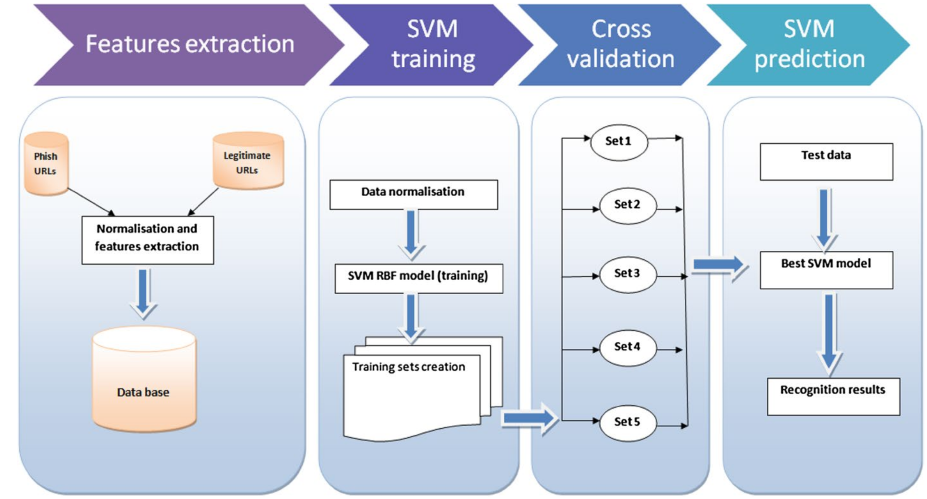

Fig. 3 Phishing detection system

However, for the similarity distance calculation algorithms we have used the Debatty java string similarity library [22]. Furthermore, we have used the Encog library [23] for all the algorithms of artificial intelligence our system needed.

Table 4 shows a fragment of our database containing successively five examples of phishing websites and four examples of legitimate websites.

Tables 5, 6, 7 and 8 show the respective error rates of each recognition algorithm tested during this work with our database plus each column in these tables represents the number of records used for each test, moreover each line of these tables represents the recognition algorithm used and the distance calculation method used, besides the other recognitions features which have already been introduced. Except for the first line of each of these tables, since we didn't input any distance calculation method, in order to measure the impact of the similarity concept on the improvement of the recognition rate. Tables 5, 6, 7 and 8 describe all the tests conceived and implemented in this study; we chose to share the results of the Tables 6, 7 and 8 despite the unsatisfactory recognition rate because they make a point about the exceptional impact of the similarity index on the recognition rate.

As shown in Table 5, we can notice that for the SVM, the best method associated with the calculation of similarity is the Hamming distance since the recognition rate is 95.80%.

In contrast, in Table 6, the method based on Bayes networks associated with the similarity calculation method based on the distance from Q-Gram ensures a recognition rate of 65.60%.

While analyzing Table 7, it can be noted that the best associated method for calculating similarity is the normalized distance Levenshtein for the algorithm based on the network Naive Bayes, with a recognition rate of 67.20%.

In Table 8, the algorithm based on Probabilistic Neural networks PNN reached a recognition rate of 51.20% free of association with any similarity calculation method, unlike other tests.

Table 4 Fragment from the database

Site	Target	Length	@	–	.	[0–9]	IP?	NL	L	JW	LCS	QG	H	State
kinsloglasswall.com	capitecbank.co.za	19	0	0	1	0	0	0.89	17	0.45	26	30	18	P
zooplaneta.sumy.ua	capitecbank.co.za	18	0	0	2	0	0	0.77	14	0.60	23	31	18	P
aussiehydrovac.staginghost.com.au	capitecbank.co.za	33	0	0	3	0	0	0.75	25	0.55	28	44	31	P
guneva.net	capitecbank.co.za	10	0	0	1	0	0	0.82	14	0.46	21	25	17	P
reg-playiing.byethost11.com	zynga.com	27	0	1	2	2	0	0.74	20	0.51	22	26	27	P
capitecbank.co.za	capitecbank.co.za	17	0	0	2	0	0	0	0	1	0	0	0	L
zynga.com	zynga.com	9	0	0	1	0	0	0	0	1	0	0	0	L
santander.co.uk	santander.co.uk	15	0	0	2	0	0	0	0	1	0	0	0	L
facebook.com	facebook.com	12	0	0	1	0	0	0	0	1	0	0	0	L

Table 5 The evolution of the error rate of the SVM related to the records count in the database

Records count	100	250	500	750	1000	1250	1500	1750	2000
SVM (%)	40	41.26	34.40	36.70	28	30.35	28.26	23.97	26
SVM—JW (%)	24	19.04	17.60	16.48	12	12.77	8	12.32	12.20
SVM—H (%)	8	11.11	5.60	2.12	4	2.55	5.60	5.02	4.20
SVM—LCS (%)	12	9.52	4.80	2.65	3.20	3.19	2.66	3.42	5.20
SVM—L (%)	0	4.76	3.20	1.06	2.80	2.23	2.93	5.93	5.60
SVM—NL (%)	20	39.68	16	11.70	16.8	15.01	15.73	14.84	13.4
SVM—QG (%)	0	7.93	2.40	2.65	4.80	5.75	3.20	4.10	5

4.20% is the lowest error rate obtained in these tests while using the complete dataset

Table 6 The evolution of the error rate of the bayesian network related to the records count in the database

Records count	100	250	500	750	1000	1250	1500	1750	2000
Bayes (%)	52	47.61	40.80	28.72	33.20	41.85	33.86	33.10	38.80
Bayes—JW (%)	64	36.50	52	29.78	38	38.01	36.80	38.12	39.20
Bayes—H (%)	4	7.93	2.40	3.72	2	38.01	36	36.98	36.80
Bayes—LCS (%)	36	6.34	6.40	2.65	0.4	34.82	35.46	36.30	39.60
Bayes—L (%)	16	6.34	5.60	2.12	7.20	0.63	6.66	35.38	37.6
Bayes—NL (%)	32	36.50	32.8	35.10	31.60	39.29	34.13	37.67	36.4
Bayes—QG (%)	24	6.34	2.40	2.12	42.4	36.42	39.20	36.98	34.4

34.4% is the lowest error rate obtained in these tests while using the complete dataset

Table 7 The evolution of the error rate of the naïve bayesian network related to the records count in the database

Records count	100	250	500	750	1000	1250	1500	1750	2000
Naive bayes (%)	52	53.96	36.8	36.70	34.40	36.74	33.60	38.12	35.8
Naive bayes—JW (%)	72	26.98	47.20	37.76	39.20	37.69	28.8	35.15	37.6
Naive bayes—H (%)	16	9.52	8	1.59	5.20	38.97	35.46	36.98	36
Naive bayes—LCS (%)	20	11.11	4.80	3.19	0.4	37.06	32.80	36.75	40.6
Naive bayes—L (%)	16	19.04	2.40	2.12	4	10.22	5.86	36.75	35
Naive bayes—NL (%)	44	50.79	40.80	27.12	36.40	37.06	39.2	25.79	32.80
Naive bayes—QG (%)	12	6.34	1.60	4.78	35.2	35.14	41.06	36.07	34.8

32.80% is the lowest error rate obtained in these tests while using the complete dataset

Table 8 The evolution of the error rate of PNN network related to the records count in the database

Records count	100	250	500	750	1000	1250	1500	1750	2000
PNN (%)	64	57.14	51.2	49.46	46	53.03	49.60	52.73	48.8
PNN—JW (%)	56	57.14	54.4	47.34	52.8	49.2	52	49.31	49
PNN—H (%)	96	84.12	92	100	99.2	96.80	97.86	99.31	98.40
PNN—LCS (%)	96	88.88	96.8	100	99.2	97.12	98.66	97.03	97.8
PNN—L (%)	92	95.23	97.6	100	98.8	100	99.73	98.40	98
PNN—NL (%)	72	49.20	64	52.65	52.4	53.03	52.53	52.05	51.80
PNN—QG (%)	100	92.06	94.40	100	98	87.85	96.80	98.17	98.40

48.8% is the lowest error rate obtained in these tests while using the complete dataset

As shown in Fig. 4 and to give more visibility to Table 5 we can notice the influence of similarity on the error rate of the recognition system, we can, therefore, note that:

1. The highest error rates are achieved during the absence of a method of calculating similarity.
2. Jaro-Winkler distance and Normalized Levenshtein distance are not optimal for this type of data despite that they improve overall the error rate.
3. Hamming distance, Q-Gram, Levenshtein and longest common subsequence have improved the error rate drastically.
4. The best recognition rates achieved in this study is 95.80% using SVM provided with the Hamming distance and several other features as input to our system.
5. Based on the results of this study, we can deduce that the substituted characters' positions between the phishing websites and their legitimate counterparts are the most important aspect of the similarity index, in order to improve the recognition rate, since the Hamming distance allowed us to reach a higher recognition rate than that obtained while using the longest common subsequence and Q-Gram distance, which does not underline the positions of the substrings and the Q-Grams. We can also infer that the computation of added characters' editions (insert, delete, replace) in the phishing websites links offered by the Levenshtein distance does not improve the recognition rate.

Implications

Affirmed by the results of our tests, we have demonstrated the potential impact of the use of the similarity distance on the detection of phishing websites. Indeed, in three tests performed on four, the introduction of the distance of similarity has

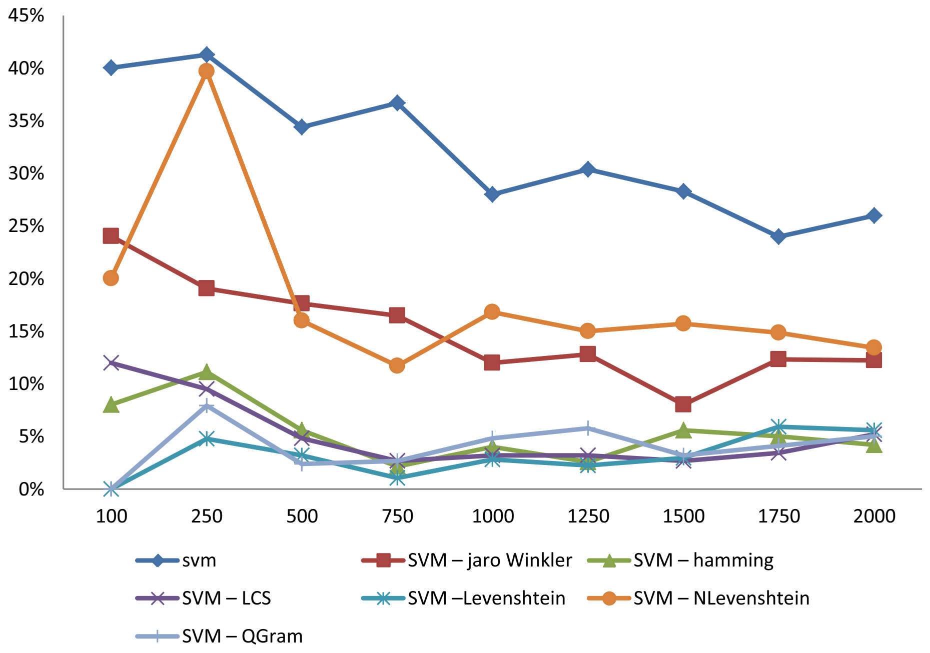

Fig. 4 Error rate based on the number of records in the SVM method compared to other methods

significantly improved the recognition rate of our detection system. In the same context, the only case where the similarity did not have a positive impact on the phishing websites recognition rate is the test with Probabilistic Neural networks that records the worst recognition rate among all of our tests. This impact is most obvious in tests performed using the SVM method since the use of the Hamming distance as one of the input characteristics of our system has improved the recognition rate of 21.8%.

We are confident that our phishing website detection system will play a key role in the war against the scourge called phishing because as shown in Table 9 it's light and more suited to the less "robust" devices such as smartphones and embedded systems since it requires only six features as an input parameter which makes it less "greedy" in terms of CPU and "memory" unlike other proposed systems. Furthermore, all characteristics used by our system are totally extracted from the URL and therefore, we do not need HTML elements of a website or to perform an image processing on the webpage of that latter to decide whether is it a phishing website or not. Besides our system does not need an HTTPS certificate to work; in other words, one bad CA wouldn't compromise the security of our system.

Conclusion

In this paper, we have presented a phishing websites detection system 100% based on the URL. Our system has been tested on a database of 2000 records formed from legitimate websites and their phishing counterparts; our system has given very satisfactory and encouraging results precisely a 95.80% recognition rate as shown by the results of the tests. The used approach in this system rests on a powerful tool of AI precisely support vector machine, provided with the Hamming distance between the phishing website and its target and five other features extracted from the URL as input. The advantage of this system is its lightness and it can be incorporated into smartphones and tablets.

We see as perspective to this work to test this system constantly on gigantic phishing websites database to improve it if this is mandatory. We will also use the methods of probabilistic prediction on the phishing websites to predict potential target website based solely on the URL of the phishing website.

Table 9 Comparison between this work and some literature relevant works

	Recognition rate (%)	Artificial intelligence	URL features	HTML features	Search engines	Image recognition	DNS look up	Data mining
This work	95.80	+	+	−	−	−	−	−
CANTINA+ [5]	92	+	+	+	+	−	−	−
Li et al. [7]	99	+	−	−	−	+	+	−
Ramesh et al. [10]	99.62	−	−	−	+	−	+	+

Additional file

Additional file 1. In the supplemental material section the entire data-set used in this study testes.

Abbreviations

URL: uniform resource locator; SVM: support vector machine; e-commerce: electronic commerce; m-commerce: mobile commerce; FBI: Federal Bureau of Investigation; HTML: hypertext markup language; PSO-SVM: particle swarm optimization support vector machine; DNS: domain name system; IP: internet protocol; TID algorithm: Target Identification algorithm; HTTPS: hypertext transfer protocol secure; NH: Number_of_Hyphens; ND: Number_of_Dots; NNC: Number_of_Numeric_Chars; IP: IP_Presence; L: the classic Levenshtein distance; NL: Normalized Levenshtein distance; JW: Jaro Winkler distance; LCS: the longest common subsequence; QG: the Q-Gram distance; H: the Hamming distance.

Authors' contributions

MZ carried out the studies, and drafted the manuscript. BO provided full guidance and revised the manuscript to high standards. Both authors read and approved the final manuscript.

Author details

[1] TSE Research Team, ENSIAS, Mohammed V University of Rabat, Rabat, Morocco. [2] FSJES Souissi, Mohammed V University of Rabat, Rabat, Morocco.

Acknowledgements

I would like to express my profound gratitude to my family for their constant support which was vital in order to finish this paper and the reviewers for their valuable comments.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The dataset used in this study was added as Additional file 1.

Funding

This work was supported by the Grant (018UM5S2014) from the Moroccan Center for Scientific and Technical Research.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 4 November 2016 Accepted: 16 May 2017
Published online: 08 June 2017

References

1. Krebs B (2014) Report on the magnitude of the business money lost to the phishing attack. http://krebsonsecurity.com/2015/08/fbi-1-2b-lost-to-business-email-scams. Accessed 8 May 2017
2. APWG (2016) The fishing activities trends' reports by the anti-phishing working group on the first quarter of 2016. https://docs.apwg.org/reports/apwg_trends_report_q1_2016.pdf. Accessed 8 May 2017
3. Microsoft (2005) Anti-phishing white paper. http://www-pc.uni-regensburg.de/systemsw/ie70/Anti-phishing_White_Paper.doc. Accessed 8 May 2017
4. Schneider F, Provos N, Moll R, Chew M, Rakowski B (2007) Phishing protection design documentation. https://wiki.mozilla.org/Phishing_Protection:_Design_Documentation. Accessed 8 May 2017
5. Xiang G, Hong J, Rose CP, Cranor L (2011) CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. ACM Trans Inf Syst Secur 14(2):21. doi:10.1145/2019599.2019606
6. Fu AY, Wenyin L, Deng X (2006) Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). IEEE Trans Dependable Secur Comput 3(4):301–311. doi:10.1109/TDSC.2006.50
7. Li Y, Chu S, Xiao R (2015) A pharming attack hybrid detection model based on IP addresses and web content. Optik-Int J Light Electron Optics 126(2):234–239. doi:10.1016/j.ijleo.2014.10.001
8. Thomas K, Grier C, Ma J, Paxson V, Song D (2011) Design and evaluation of a real-time URL spam filtering service. In: proceedings of the 32nd IEEE symposium on security & privacy, California, 22–25 May 2011, p. 447–462
9. Jeeva SC, Rajsingh EB (2016) Intelligent phishing url detection using association rule mining. Human-centric Comput Inf Sci 6:10. doi:10.1186/s13673-016-0064-3
10. Ramesh G, Krishnamurthi I, Kumar KSS (2014) An efficacious method for detecting phishing webpages through target domain identification. Decis Support Syst 61:12–22. doi:10.1016/j.dss.2014.01.002
11. Huang C-Y, Ma S-P, Chen K-T (2011) Using one-time passwords to prevent password phishing attacks. J Netw Comput Appl 34(4):1292–1301
12. Yue C, Wang H (2010) BogusBiter: a transparent protection against phishing attacks. ACM Trans Int Technol 10(2):1–31. doi:10.1145/1754393.1754395
13. Phishtank phishing websites database. http://data.phishtank.com/data/online-valid.csv. Accessed 8 May 2017
14. The top accessed 500 websites on the web. http://www.alexa.com/topsites. Accessed 8 May 2017

15. Levenshtein V (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10(8):707–710
16. Jaro MA (1995) Probabilistic linkage of large public health data files. Stat Med 14(5–7):491–498
17. Yujian L, Bo L (2007) A normalized levenshtein distance metric. IEEE Trans Pattern Anal Mach Intell 29(6):1091–1095. doi:10.1109/TPAMI.2007.1078
18. Apostolico A, Guerra C (1987) The longest common subsequence problem revisited. Algorithmica 2(1):315–336. doi:10.1007/BF01840365
19. Ukkonen E (1992) Approximate string-matching with q-grams and maximal matches. Theor Comput Sci 92(1):191–211. doi:10.1016/0304-3975(92)90143-4
20. Hamming R (1950) Error-detecting and error-correcting codes. Bell Syst Tech J 29(2):147–160
21. Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag New York, Inc, New York
22. Debatty T (2015) The tdebatty java string similarity library. https://github.com/tdebatty/java-string-similarity. Accessed 8 May 2017
23. Heaton J (2015) Encog: library of interchangeable machine learning models for java and C#. J Mach Learn Res 16:1243–1247