Journal of
Computational and Theoretical Nanoscience
Vol. 16, 1–7, 2019

Phish Detect-Real Time Phish Detecting Browser Extension

Shiva Nandhini*, K. Riya, P. Divya, Sahil Khatri, and V. Devi Subadra

SRM Institute of Science & Technology, Kattankulathur 603203, India

Now a days, hacking has become a trend, where the personal information or user data including login credentials, credit card numbers, such as transaction from our bank accounts, key information from government office, defense etc., which threatens the privacy and property security of netizens in wireless communication. This is being done by creating a shadow website which has similar looks and semantics of the legitimate website. So as to overcome these kind of circumstances, the concept of Phishing is introduced. The phishing is a type of social engineering attack to detect false URL's. So, the paper introduces a new phishing website "PhishDetect" to check for the phishing or fake sites. The paper proposes a detection technique of phishing websites based on checking URL of webpages. The detected attacks are reported for prevention of hacking.

Keywords: JS, SQL, CSS, Word-to-Vec, Phishing Sites, Machine Learning, Browser Extensions.

RESEARCH ARTICLE

1. INTRODUCTION

The importance and need of not being get hacked is more important in today's world. The hacking is a process of stealing of information from users. The hacking does not only lead to the personal loss of information, but it may also lead to the loss of country data by misusing the information by terrorist which helps them for attacking on individuals and government properties. So as to avoid even a little chance of getting hacked can be prevented by using a website which helps in detecting the phishing sites. The PhishDetect is a website which detects phishing sites, using URL's. The series of semantic features through word2vec using machine learning techniques is done by which it creates a more robust phishing detection model. A deep learning approach was proposed to get distributed word representation of words through training by a language model. A browser addon is made to make real time detection of phishing websites. Natural Language Processing (NLP) helps in convert human language into a formal representation which is easy for computers to manipulate. Distributed word representation are also called word embeddings which can have both semantic and syntactic information of words. The training is based on weight-sharing which is an instance of multitask learning. All the tasks such as part-of-speech tagging, chunking, named-entity recognition, learning a language are integrated into a single system, all the tasks except the language model are supervised tasks with labeled training data. The language model is trained in an unsupervised fashion [1].

Recent years have witnessed a huge advancement in technology due to which the internet services are in a high demand. Use of mobile devices have been increased a lot due to which many websites are being built for desktop and mobile phones separately. However, the phisher does not necessarily built different phishing websites, because he does not want to spend much of his time for better user experience. Detection of phishing websites is done through device detection. In the experiment, whitelist and SVM classifier is used to detect the phishing website. If the URL is same as in whitelist it will be considered as non-phishing, if there is slight difference then SVM classifier is used to figure out which makes predictions based on common phishing features [2].

A method is proposed which uses a logo image to determine the identity consistency between two websites. A Consistent identity indicates a legitimate website and inconsistent identity indicates a phishing website. It is done using two methods namely logo extraction and identity verification. The first process will detect and extract the logo image from all the downloaded image resources of a webpage. The right logo image is detected using machine learning techniques. Google image search based on the extracted logo is used to retrieve the portrayed identity. Since the relationship between the logo and domain name is exclusive, it is reasonable to treat the domain name

[*] Author to whom correspondence should be addressed.

1546-1955/2019/16/001/007 doi:10.1166/jctn.2019.8423

as the identity. Hence, a comparison between the domain name returned by Google with the one from the query website will enable us to differentiate a phishing from a legitimate website. The conducted experiments show reliable and promising results. This proves the effectiveness and feasibility of using a graphical element such as a logo to detect a phishing website [3].

Phishing, also known as brand spoofing, which seriously threatens the web security. After analysing lots of phishing data of PhishTank and Anti-Phishing. The statistics show that almost every phishing sites contain at least one brand entity. Favicon, logo and copyright notice are the most important brand identities of company sites, which are widely used by phishers to trick the users. Favicon files can be detected by parsing the Web page source code. Redirection, incoming links and Domain Name System (DNS) information are further extracted to discriminate the sites with branding rights from phishing sites [4].

Now a days, usage of the Uniform Resource Locator (URL) as a vector for influencing internet users have become common. We are going to focus on a complementary technique—lightweight real-time classification of the URL itself to predict whether the associated site is malicious or not. We use various lexical and host-based features of the URL for classification, but, by excluding Web page content. As a result of which, security researchers have developed various systems to protect users from making wrong choices. The phishing websites are kept under "black listing," for representing them as the phishing websites.

Using this, the third-party service compiles the names of "phishing" Web sites and reports the list to their users. So, there is a chance of clicking on a malicious URL by the user before it appears on a blacklist, but later on he can't repeat the same mistake, because after subscribing to this upcoming website he would be able to check all the phishing sites. By the end, our papers primary contribution is a successful application of online learning algorithms to the problem of predicting such malicious URLs. However, practically the online methods are far better suited in nature for solving problems. By comparing the classical and modern online learning algorithms and by finding the Confidence-Weighted algorithm, we achieve accuracies up to 99% over a balanced data set [5].

The two main findings of our preliminary work include: (i) Phishing URLs, (ii) Domain names which have very different lengths compared to other URLs and the name of the domains in the Internet. Even the character of the frequency, where the phishing domain names is significantly different from English when the DMOZ URLs and domains follow the English letter character frequency very closely. Further, 75% of the phishing URLs contain the name of the brand they targeted. Free Web hosting services as well as URL-aliasing services, such as, Tiny URL are misused by the phishers. This points to the need to better scrutinize the users of such services [6].

We propose a machine learning based approach to classifying phishing webpages by using the information available on URLs, hosting servers, and page contents. We treat the problem of detecting phishing webpages as a binary classification problem where we classify phishing webpages from legitimate non phishing ones. We first run a number of scripts to collect our phishing and non-phishing URLs, automatically fetch their page contents and create our data sets. Our next batch of scripts then extracts a number of features by employing various publicly available resources in order to classify the instances into their corresponding classes. We then apply machine learning algorithms to build models from training data, which is comprised of pairs of feature assignments and class labels. Separate sets of test data are then supplied to the models, and the predicted class of the data instance (phishing or non-phishing) is compared to the actual class of the data to compute the accuracy and various other performance measures of the classification models [7].

One reason that the phishers' tactics are increasingly effective and rapidly is: Instead of asking directly from users for bank account information in scam e-mails or on the Web sites, cybercriminals are using hidden/shadowed malicious software's which are downloaded to users' desktops that monitors their online activities and records bank codes, Litan says. That means that the users don't need to reveal financial information themselves, only by clicking on a link that directs them to a malware-infected Web page [8].

2. RELATED WORKS

Phishing website detector is possible. When the method falls under these three categories. They are:- Blacklist-based methods, heuristic method and machine learning-based methods.

2.1. Blacklist-Based Methods

This method approaches by using the URL's Address of the website can be able to detect the phishing website by comparing it with the blacklist, which will contain the database list of the illegitimate URL's. So, by comparing the domain's name with the illegitimate website, can be able to verify whether it is free from phishing (white list) or not.

2.2. Heuristic Method

It will even use the features and classifiers for the evaluating of the features it provides the classifiers for the detection of the website. It can able to detect the temporary and the permanent websites.

Cantina: Cantina which comes under heuristic method. A novel substance based approach for recognizing phishing sites. Bar takes Hearty Hyperlinks, an thought for defeating page not discovered issues utilizing the outstanding Term Recurrence/Reverse Archive Recurrence

(TF-IDF) calculation, and applies it to hostile to phishing. Our execution of Bar, and some basic heuristics that can be connected to lessen false positives. Likewise, displayed an assessment of Cantina, demonstrating that the unadulterated TF-IDF approach can get around 97% phishing locales with around 6% false positives, and after joining some straightforward heuristics can get around 90% of phishing locales with just 1% false positives.

Pilfer: The another strategy for distinguishing these vindictive messages called Pilfer. By joining highlights particularly intended to feature the misleading strategies used to trick clients, we can precisely arrange more than 92% of phishing messages, while keeping up a false positive rate on the request of 0.1%. These outcomes are acquired on a dataset of roughly 860 phishing messages and 6950 non-phishing messages. The precision of Appropriate on this dataset is essentially superior to that of Spam Professional killer, a broadly utilized spam channel.

Malicious Website URLs: A way to deal with this issue in light of mechanized URL arrangement, utilizing measurable strategies to find the obvious lexical and host-based properties of pernicious Site URLs. These strategies can learn exceedingly prescient models by separating and consequently examining a huge number of highlights possibly characteristic of suspicious URLs. The subsequent classifiers get 95–99% precision, recognizing substantial quantities of pernicious Sites from their URLs, with just humble false positives.

Page Rank: By utilizing the PageRank esteem and different highlights to characterize phishing locales from ordinary destinations. The gathered a dataset of 100 out of the W3C models to assess the security of the sites, and check each character in the website page source code, in the event that be able to discover a phishing character, and we will diminish from the underlying secure weight. At last we compute the security rate in view of the last weight, the high rate demonstrates secure site and others shows the site is destined to be a phishing site. By checking the two site page source codes for true blue and phishing sites and think about the security rates between them, discover the phishing site is less security rate than the true blue site; our approach can distinguish the phishing site in view of checking phishing qualities in the site page source code.

Behaviour Based Detection: A ideal way to deal with identify phishing sites in light of examination of clients' online practices—i.e., the sites clients have visited, and the information clients have submitted to those sites. Such client practices can't be controlled uninhibitedly by assailants; discovery in light of those information can accomplish high exactness, as well as is on a very basic level flexible against changing misdirection strategies.

Lexical Analysis: A lexical URL examination (LUA) system to improve the characterization exactness of hostile to phishing email channels. In spite of the fact that the LUA include is fundamentally engaged to order phishing sites, it turned out to be viable to group email messages because of the way that most phishing email messages contain URLs. As indicated by the execution assessment, the LUA highlight turned out to be successful in improving the classifier's exactness in all highlights subsets.

Detecting Webpage Source Code: A phishing recognition approach in light of checking the site page source code, can able to remove some phishing attributes, phishing destinations and 100 true blue locales for our utilization. By utilizing this Google PageRank system 98% of the destinations are accurately ordered, demonstrating just 0.02 false positive rate and 0.02 false negative rate.

3. DISADVANTAGES WITH THE EXISTING SYSTEM

The drawback of the black list-based method is it can't able to distinguish between new phishing website and the legitimate website.

In the heuristic method is more efficient than the blacklist-based method in detecting the phishing website but, even it has its own drawback in that heuristic method cannot able to check and access the website directly with the real time websites, only through copy-paste the website's URL. It is very difficult to implement also. Its the major drawback in this evolutionally growing global world.

4. METHODOLOGY

Detection of phishing is done using machine learning and real time phishing detection is provided using browser extension. Website of any URL visited by user is fetched and compared with the whitelist of websites, so as to reduce the time taken for detection. White list consists of URL of all the websites which have been declared as non-phishing. If the URL of the website is matched with the existing list then no processing is required, otherwise the website undergoes examination by the classifier. The URL, lexical features and semantic along with favicons and icon are taken into consideration, they are translated into the format on which the classifier can work on.

4.1. Features Used

There exists a number of features for detection of phishing websites we have used semantic and statistical features.

(A) Semantic features:

Word Embeddings: Word Embeddings represents both semantic, synatic representation of words, which have been used in many Natural Language Processing. The common method used for word embeddings generation is training a language model using neural networks. Each word is associated with a continuous vector representation, each represent the grammatical characteristics of a word.

(B) Statistical Features

(a) Domain name age:

Attackers use phishing websites only for a certain period of time to avoid being caught. We can examine the domain age of a particular website using WHOIS query along with URL, it provides the information regarding name of person domain registered to, place, expiry date etc.

(b) Number of Subdomains

Attackers make use of subdomains to make a website look original, if a site has subdomains it will have many dots in its URL, this is also used to flag a website as phishing or not.

(c) Presence of Form tag:

Most phishing websites usually use a form to collect data from the user, the form may be maliciously submitted elsewhere rather than the authorized domain.

It is checked whether the information provided is going to legitimate or being directed elsewhere.

(d) Copyright Features:

The copyright section of a legitimate website contains the copyright information which is usually missing on fake websites.

(e) ICP Number

The People's Republic of telecommunications and information services business license (ICP) is issued by communications management department, it is not possible for a phishing website to have a ICP Number.

First it was checked if a website possessed a ICP number or not and if it did the domain name was matched with the legitimate website to check for phishing.

4.2. Obtaining Word Embeddings

Word embeddings are obtained using the following methods:

(a) Segmentation

After extracting html text from the webpages the segmentation was done using a segmentation tool named Word Segment, it is an Apache2 licensed module used for english word segmentation.

(b) Language model training

Training a language model is the important process, the web page texts processed by the tool were used to train the language model. Word2Vec of CBOW model was selected to train the model. Word embeddings are used so as to place the similar words close to each other in feature space, which makes it possible to calculate the similarity between the words by finding the distance between them.

(c) Semantic Representation

The final step is semantic representation of legitimate and phishing text

- Removal of stop words:

  Stop words are generally considered irrelevant and thus are removed which makes it easier for the further process.

- Calculation of mean of word embeddings Vector of a given page is calculated by using the formula (1)

$$d_j = \frac{1}{n_j} \sum_{j-1}^{n_j} w_{ij} \tag{1}$$

where d_i—Semantic representation of text, n_i—Number of words, w_{ij} word embedding of j-th word in i-th text i—text.

- Calculation of TF-IDF.

TF-IDF (term frequency-inverse document frequency) is a numerical value which helps in identifying the importance of a particular word in a ext. TF-IDF was used as a weight and corresponding text vector was calculated using formula (2).

$$d_i = \frac{1}{\sum_{j-1}^{m_j} tfidf_{ij}} \sum_{j-1}^{m_j} tfidf_{ij} \cdot w_{ij} \tag{2}$$

w_{ij} word embedding of j-th word in i-th text, *Tfidfi i -* $tf\text{-}idf$ value of the j-th word in i-th text.

4.3. Browser Extension

A browser extension has been made which is used for real time phishing detection, whenever the user visits any website he gets to know that the website is safe or not. As the user gets to know about the genuinity of the website in real time this method is very effective and hence tends to reduce phishing attacks.

5. ARCHITECTURE DIAGRAM

Data model is trained using data set including visual feature extraction, URL detection and semantic feature extraction, the training model detects the phishing. The user can check whether a website is phishing or not using a website and browser extension. Database from phishtank is also used which makes detecting process faster, phishtank has the database of all the phishing websites which have already been identified. If match is found then it is directly termed as phishing website, similarly if match is found in Whitelist it is termed as non-phishing. If there is no record of website the data model identifies whether it is phishing or not.

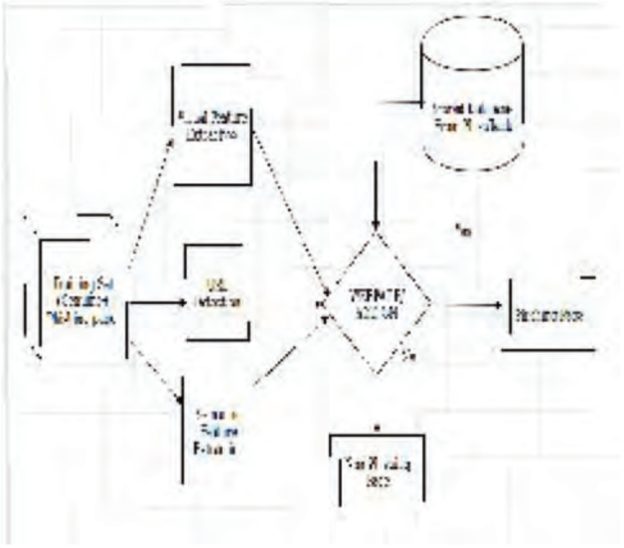


Fig. 1. Diagram representing the architecture diagram.

6. EXPERIMENTAL RESULTS AND ANALYSIS

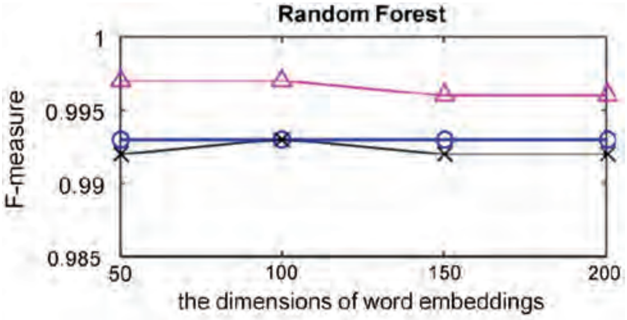
6.1. Dataset

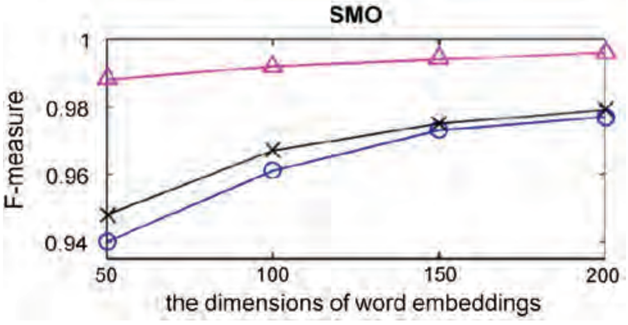
To verify the validity of the extracted features and the models constructed in this study, we constructed a complex, extensive and practical web page dataset, including legitimate and phishing web pages. The legitimate URLs and websites in the dataset were obtained from PhishTank and DirectIndustry web guides, collected by retrieve brand names in the search engine, and domain names randomly selected from DNS resolution logs. Based on these URLs, 26786 HTMLs were crawled through simulating PC and mobile phone respectively and checked manually, including brand sites and a number of sites easily confused (in terms of contents, site titles, and domain names, etc.). The phishing data was obtained from PhishTank, reported by users, totalling 26k items able to be visited after removing part of template duplication data. In total, many brands were involved, payment sites, covering bank sites, e-commerce sites, securities sites, mobile service sites and many more. On the bases of constructed dataset, we compared statistical features, the influence of the dimension of word embeddings and investigated different feature selection types (the semantic features, the fusion features and the statistical features) on the performance of phishing detection.

6.2. The Effect of the Dimension of Word Embeddings

Word embeddings with different dimensions were trained, and the numbers of dimensions were 50, 100, 150, and 200 respectively. Then mean of word embeddings, fusion of mean of word embeddings and average weighted of word embeddings and statistical features were employed to train phishing detection models with the above mentioned method's algorithms like SMO algorithms and Random Forest. We trained the models trained with different algorithms and word embeddings with different dimensions were evaluated by F-measure values, and 10-fold cross validation of the models which were calculated as the performance measure of phishing detection. The closer to 1 F-measure is, the better the model performs.

Relationships between F-measure value and the dimensions of the word embeddings for each algorithm were illustrated in figure.





As shown in figure, the obtained models except trained by SMO, presented excellent performance on phishing detection with F-measure values over 0.985, and the performance of the models showed little improvement with the increase of the dimension of word embeddings. However, the computing resources occupied increased dramatically with the increase of the dimension of word embeddings. That's why, the training of phishing detection models is optimal for 50-dimensional word embeddings.

6.3.

We proposed the comparison of statistical features for the statistical features, we analysed the information gain ratio of each single feature, and conducted comparative experiment to find the result of each class of statistical features on phishing detection. The information gain ratio is shown in Figure 2(a). The greater the information gain ratio of a feature, the greater the chances of the feature to reduce the entropy of samples. In Figure 2(a), the information gain ratio of the feature "whether the validity of domain name is less than one year" (f8) is the largest and the feature "to check whether it is copyright or not" (f4) is the small. Figure 2(b) shows that the accuracies of the 5 classes of statistical features using SMO algorithm are all over 50%,

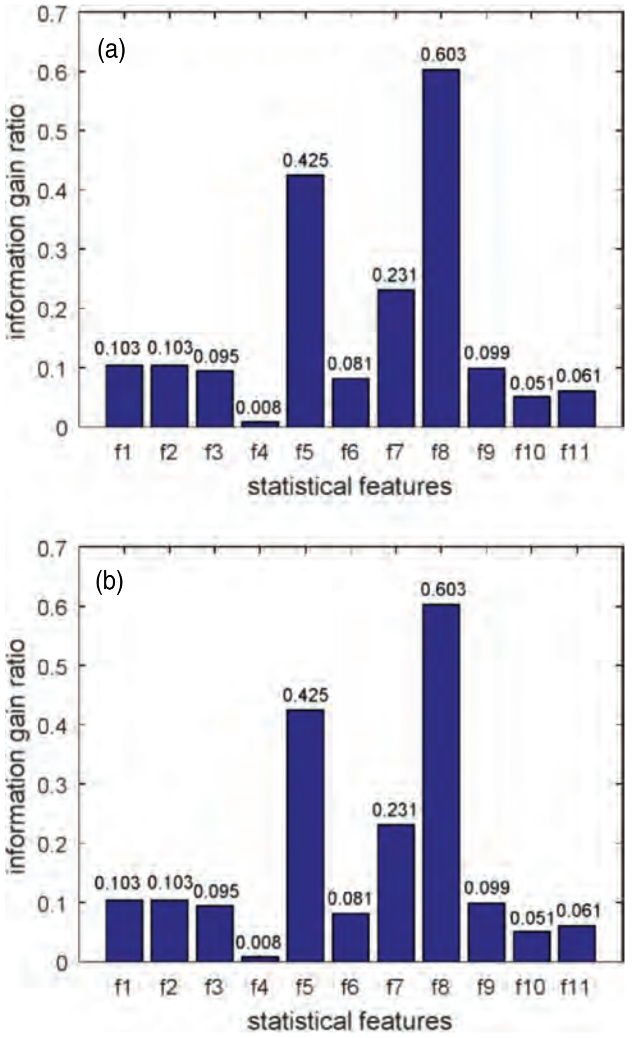
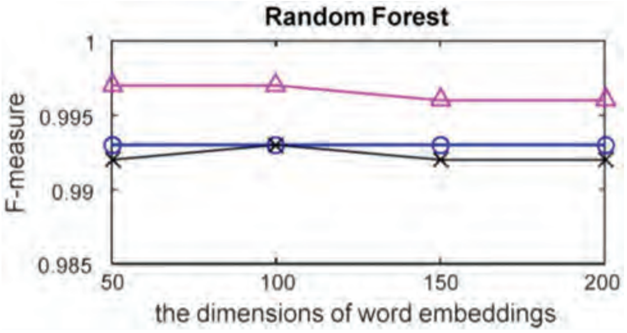


Fig. 2. (a) Single features information gain ratio. (b) The accuracy of different classes of the extracted statistical features.

which demonstrates that all of these types of statistical features have positive contributions to phishing detection. The accuracies of the license counterfeiting features and domain name aging features are the top two, which are 0.9271 and 0.8635 respectively, indicating that two types of statistical features have stronger ability to recognize phishing sites.



6.4.

In this study, four feature spaces were respectively tested by the four machine learning algorithms, to evaluate the result of different features on phishing detection. The four features were mean word embeddings features, weighted word embeddings features, statistical features (the linear fusion of stealing features, license counterfeiting features, copyright counterfeiting features, domain name aging features and consistency features described) and the fusion of mean word embeddings features and statistical features, respectively. The phishing detection performance measures including precision rate (P), recall rate (R), F-measure, false positive (FP) rate, error rate and ROC area, and the average measures of 10-fold cross validation measures

Table I. The performances of the detection models based on difference features spaces.

		Performance measures					
Algorithm	Features	P	R	F-measure	FP rate	Error rate	ROC area
AdaBoost	Word embeddings (mean)	0.991	0.991	0.991	0.014	0.0088	0.998
	Word embeddings (weighted)	0.992	0.992	0.992	0.014	0.0085	0.998
	Statistical features ($t1 \cup t2 \cup t3 \cup t4 \cup t5$)	0.991	0.991	0.991	0.012	0.0089	0.998
	Word embeddings (mean)∪statistical features	**0.998**	**0.998**	**0.998**	**0.003**	**0.0015**	**0.999**
Bagging	Word embeddings (mean)	0.990	0.990	0.990	0.016	0.010	0.998
	Word embeddings (weighted)	0.990	0.990	0.990	0.016	0.0097	0.999
	Statistical features	0.983	0.983	0.983	0.023	0.0167	0.996
	Word embeddings (mean)∪statistical features	**0.996**	**0.996**	**0.996**	**0.007**	**0.0040**	**0.999**
Random forest	Word embeddings (mean)	0.993	0.993	0.993	0.013	0.0069	0.999
	Word embeddings (weighted)	0.992	0.992	0.992	0.015	0.0080	0.999
	Statistical features	0.991	0.991	0.991	0.013	0.0092	0.997
	Word embeddings (mean)∪statistical features	**0.996**	**0.996**	**0.996**	**0.008**	**0.0042**	**1**
SMO	Word embeddings (mean)	0.977	0.977	0.977	0.03	0.0225	0.974
	Word embeddings (weighted)	0.979	0.979	0.979	0.027	0.0213	0.976
	Statistical features	0.954	0.953	0.954	0.052	0.0466	0.95
	Word embeddings (mean)∪statistical features	**0.996**	**0.996**	**0.996**	**0.006**	**0.0044**	**0.995**

were calculated. According to the data in Table I, following conclusions are summarized.

- The models trained by different algorithms on the four feature spaces in this paper performed well in all performance measures. In general, the detection performance of the learnt models only based on word embeddings features is comparable with using statistical features, while the fusion models using the fusion of word embeddings features and other scale statistical features attains the best performance.
- Fusion models have significant advantages in F-measure, false positive rate and error rate. The F-measure of fusion models are all more than or equal 1069 to 0.996, even reaching 0.998, while the highest F-measure of other models is 0.993. The false rates of fusion models are either equal to or less than 0.008, 0.003 for the best, while the false positive rates of other models are all more than or equal to 0.012, even 0.052 for the SMO model based on the statistical features. And the error rates of fusion models are either equal to or less than 0.0044, even 0.0015 for the AdaBoost fusion model, which is much lower than the other models.
- As to the stability of the constructed model, the performance of word embeddings based detection models is pretty stable, in which the fusion models have the best performances. However, the performances of other scale statistical features based models are greatly influenced by the machine learning algorithm used. The stability of the statistical features based model trained by SMO algorithm are obviously inferior to other models.

7. FUTURE WORK AND CONCLUSION

In future, the phishing will have a great importance, as the demand for hacking increases. The paper consists of few features such as the sematic features and statistical features. The present paper consist of an addon which is accessible now only from "Google chrome. So in future it can be further implemented in other web browsers. "Prevention is better can cure" in future we can make maximum of the sites "hack free," which simultaneously helps in decreasing the hacking process, by providing more security to sites. Using the data analytics, machine learning, deep learning the data security will be increased. Thereby, increasing the data security the efficiency of the site also increases and decreases the hacking process. The users will be able to be in safer side, by detecting the phishing sites more efficiently and easily using Phish Detect extension.

References

1. Collobert, R. and Weston, J., **2008**. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proceedings of the 25th International Conference on Machine Learning*, July; ACM. pp.160–167.
2. Lin, I.C., Chi, Y.L., Chuang, H.C. and Hwang, M.S., **2016**. The novel features for phishing based on user device detection. *JCP, 11*(2), pp.109–115.
3. Chiew, K.L., Chang, E.H. and Tiong, W.K., **2015**. Utilisation of website logo for phishing detection. *Computers & Security*, *54*, pp.16–26.
4. Geng, G.G., Lee, X.D. and Zhang, Y.M., **2015**. Combating phishing attacks via brand identity and authorization features. *Security and Communication Networks*, *8*(6), pp.888–898.
5. Ma, J., Saul, L.K., Savage, S. and Voelker, G.M., **2009**. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, June; ACM. pp.1245–1254.
6. Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C., **2003**. A neural probabilistic language model. *Journal of Machine Learning Research*, *3*(Feb), pp.1137–1155.
7. https://www.forbes.com/2007/12/27/phishing-hacking-virus-tech-security cx_ag_1228phish.html#16ce02a53f25.

Received: 28 October 2018. Accepted: 8 November 2018.