Phishing Detection on Twitter Streams

Se Yeong Jeong, Yun Sing Koh(✉), and Gillian Dobbie

Department of Computer Science, The University of Auckland,
Auckland, New Zealand
sjeo017@aucklanduni.ac.nz,{ykoh,gill}@cs.auckland.ac.nz

Abstract. With the prevalence of cutting-edge technology, the social media network is gaining popularity and is becoming a worldwide phenomenon. Twitter is one of the most widely used social media sites, with over 500 million users all around the world. Along with its rapidly growing number of users, it has also attracted unwanted users such as scammers, spammers and phishers. Research has already been conducted to prevent such issues using network or contextual features with supervised learning. However, these methods are not robust to changes, such as temporal changes or changes in phishing trends. Current techniques also use additional network information. However, these techniques cannot be used before spammers form a particular number of user relationships. We propose an unsupervised technique that detects phishing in Twitter using a 2-phase unsupervised learning algorithm called PDT (Phishing Detector for Twitter). From the experiments we show that our technique has high accuracy ranging between 0.88 and 0.99.

Keywords: Phishing Detector · Twitter stream · DBSCAN

1 Introduction

As the user base of Twitter steadily grows into the millions, real time search systems and different types of mining tools are emerging to enable people to track events and news on Twitter. These services are appealing mechanisms to ease the spread of news and allow users to discuss events and post their status, but opens up opportunities for new forms of spam and cybercrime. Twitter has become a target for spammers to disseminate their target messages. Spammers post tweets containing typical words of a trending topic and URLs, usually obfuscated by URL shorteners that lead users to completely unrelated websites.

Phishing is the fraudulent attempt to obtain sensitive information such as usernames, passwords, personal details and banking details often for malicious reasons under the disguise of a trustworthy entity in an electronic community. Traditionally, phishing emails contain links that take advantage of a user's trust. Phishing sites are usually almost identical imitations of genuine ones, taking advantage of average users to obtain private information, generally financial details. With an increased popularity in social media networks, links to phishing sites are commonly found on these platforms. These links are often masked in

H. Cao et al. (Eds.): PAKDD 2016 Workshops, LNAI 9794, pp. 141–153, 2016.
DOI: 10.1007/978-3-319-42996-0_12

shortened URLs to hide true URLs. This category of spam can jeopardise and de-value real time search services unless an efficient and accurate automated mechanism to detect phishers is found.

Although the research community and industry have been developing techniques to identify phishing attacks through other media [6,8–10] such as email and instant messaging, there is very little research that provides a deeper understanding of phishing in online social media. Moreover these phishers are sophisticated and adaptable to game the system with fast evolving content and network patterns. Phishers continually change their phishing behaviour patterns to avoid being detected. It is challenging for existing anti-phishing systems to quickly respond to newly emerging patterns for effective phishing detection. Moreover relying on the network connection information means that the phishers would have to have been active over a period to build up the connections. A good anti-phishing detection algorithm should be able to detect phishing as efficiently and early as possible.

We proposed a two phase approach called Phishing Detector for Twitter (PDT), that combines a density based clustering algorithm, DBSCAN, with DerTIA algorithm used in detecting attacks on recommender systems. We adapted both these approaches to detect phishing on Twitter. The main contributions of this paper are outlined as follows: (1) Introduced a phishing detection algorithm, PDT, an unsupervised learning approach, which does not rely on social influence (social network connections); (2) Described a systematic feature selection and analysis process.

The remainder of this paper is structured as follows. Section 2 discusses various methods researchers have designed to confront this problem of detecting phishing or spam in Twitter and other media. Then in Sect. 3, we review the outline of our technique. Section 4 details the data collection methodology of our research. Section 5 discusses selected features, performs analysis on features from collected samples. Section 3 details our unsupervised technique to detect phishing in Twitter. In Sect. 7, we review experimental setups and results on several data sets. Lastly, Sect. 8 concludes the paper.

2 Related Work

Phishing has been found in various traditional web applications such as emails, websites, blogs, forums and social media networks. Numerous preventative methods have been developed to fight against phishing.

List-Based Techniques. The list-based anti-phishing mechanism is a technique commonly used at low cost. Its strength comes from speed and simplicity. Classifying requires a simple lookup on the maintained database. Blacklists [14] are built into modern web browsers. A major drawback of the list-based mechanisms is that the accuracy is highly dependent on the completeness of the list. It takes time and effort to maintain the lists. Google uses automated proprietary algorithms to maintain a list of fraud websites whereas PhishTank [1] relies on contributions from online communities.

Machine Learning Based Techniques. With increases in computing power, phishing detections involving machine learning has emerged [2]. These approaches utilise one or more characteristics found on a site and build rules to detect phishing. Pre-labelled samples have a pivotal role in buiding a classifier. Garera *et al.* [9] proposed a technique that uses structure of URL in conjunction with logistic regression classification and Google PageRank in order to determine if a URL is legitimate or phishing.

The following two techniques employ a visual cue in detection. GoldPhish by Dunlop *et al.* [7] implements an unusual classification approach that takes a screenshot of a target website and comparies it with a genuine one to find any discrete differences. In addition to the visual comparison, the classifier considers the extracted text from optical character recognition on the screenshot in the judgement. Zhang *et al.* [18] handled visual content differently. The system first takes a screenshot of the page in question and generates a unique signature from the captured image then the image is labelled by Visual Similarity Assessment (Earth Mover's Distance). At the same time, the system extracts textual information from processed content. The textual features are then classified using Naïve Bayes' Rule and combined with labelled image features. The classifications are evaluated by a statistical model to determine the final label. Cantina+ [16] is another feature based approach that detects phishing. It makes use of features found in DOM, search engines and third party services with machine learning techniques. The accompanying two novel filters are used to help reduce incorrectly labelled data and accomplish runtime speedup.

Phishing Detection Techniques for Twitter. There is no denying that social media networks have become the main target for spammers due to their increase in popularity in today's world. Yardi *et al.* [17] studied spam in Twitter using network and temporal properties. Machine learning algorithms are incorporated in these techniques in order to uncover patterns exploited by spammers. In [15], the authors proposed a method that utilises graph-based and content-based features for Naïve Bayesian classifiers to distinguish the suspicious behaviors from normal Tweets. CAT [4] and the proposed method in [5] also use classification techniques to detect spammers. While the previously mentioned studies see spam detection as a classification problem, Miller *et al.* [13] viewed it as an anomaly detection problem. Two data stream algorithms, DenStream and StreamKM++, were used to facilitate spam identification. As opposed to spam detection on Twitter, there are a few studies carried out on phishing detection on Twitter. PhishAri [3] is a system that detects malicious URLs in tweets using URL-based, WHOIS-based, user-based and network-based features using a random forest classification algorithm. Warningbird [11] is another detection system by Lee *et al.* Unlike other conventional classifiers which are built on Twitter-based and URL-based features, Warningbird relies on the correlations of URL redirect chains that share the same redirection servers.

Current techniques suffer from similar limitations which include (1) difficulty in detecting phishers before they are sufficient inter-user relationships (or network information); (2) lack of robustness to changes within phishing trends; (3) timeliness and a significant effort to maintain the completeness in List-based techniques.

3 Overview: Phishing Detector for Twitter (PDT)

Given the ever-changing nature of the Twitter stream and behaviour of phishers, we postulate that unsupervised learning techniques could be a better way to detect phishing. This section details unsupervised learning algorithms and the cluster classifier that are used to identify phishing tweets. We propose a new technique called Phishing Detector for Twitter (PDT). Our approach has two phases. In Phase 1 we used DBSCAN to determine legitimate and phishing tweets. However as DBSCAN is a strict approach there are clusters with data points that are difficult to classify as legitimate or phishing. These indeterminate tweets (data points) are then passed into Phase 2. By using this two phase approach we are able to increase the accuracy of phishing detection significantly.

In the following sections we will discuss the data collection methodology, along side feature selection and the analysis we carried out. We then discuss our PDT approach in detail.

4 Data Collection Methodology

In this section, we describe the method of data collection and how each sample was labelled for this research.

Crawling Tweets. For this study, tweets were collected using the Twitter Public Streaming API. The API offers samples of public data flowing through Twitter around the world in real time. After a sample dataset was completed, tweets that did not contain URLs were removed from the dataset since they are irrelevant to the study. On October 4th, 2014, we collected $25,350$ tweets and of those, $5,151$ with URLs were used in this research as the initial experimental set. We later crawled additional tweets as validation sets.

Labeling Samples as Phishing or Legitimate. Despite having an unsupervised learning approach it was necessary to have suitable ground truth datasets to validate our results. Thus we annotated the collected tweets as phishing or legitimate by utilising two phishing blacklists from Google Safe Browsing API and Twitter. If a user is suspended, we have labelled this user as a spammer, therefore all of the URLs in his tweets were marked as phishing sites. Due to delays in Twitters phishing detection algorithm, we have waited for seven days and checked URLs against the databases to label the collected messages.

5 Feature Selection and Analysis

Previous studies on email phishing show that some features of URLs and email can be used to determine if a URL is malicious. In terms of Twitter, it lacks some of the features that emails hold, however, they may be substituted by other metadata that only a tweet embeds.

User Features. Each Twitter user has extra information that can be utilised to identify phishing tweets along with other features. We identified seven user features and they are shown in Table 1.

Table 1. Identified User Features

Feature	Type	Description
Follower count	numeric	Number of Twitter users who follows the user
Following count	numeric	Number of Twitter users who are followed by the user
Age of account	numeric	Number of days since creation of account
Favourites count	numeric	Number of tweets that the user marked as favourite
Lists count	numeric	Number of lists that the user was subscribed to
Presence of description	nominal	Existence of profile description
Verified user flag	nominal	Whether the user's account was verified by Twitter. Verification is used to establish authenticity of identities of key individuals and brands on Twitter

Tweet Features. Malicious tweets change their attributes over time to get higher visibility in the global ecosystem. Such tweets try to gain attraction by including keywords of trending topics (often by using hashtags), mentioning popular users (denoted by @) and following other active users. They include such tokens to increase their visibility to users who use Twitter's search facility or external search engines. By mentioning genuine users randomly, phishing tweets can make themselves visible to those users. Additional metadata of a tweet can be retrieved via a Twitter API. This includes, but is not limited to, retweet counts, favourite counts and a Boolean flag a user may set to indicate presence of possible sensitive information. We identified eight tweet features shown in Table 2.

Table 2. Identified Tweet Features

Feature	Type	Description
Message length	numeric	Number of characters in a tweet
Retweet count	numeric	Number of times a tweet was retweeted by other users
Favourites count	numeric	Number of times a tweet was marked as favourite by other users
URL count	numeric	Number of URLs found in a tweet
Hashtag count	numeric	Number of hashtags (#keyword) found in a tweet
Mention count	numeric	Number of user mentions (@user) found in a tweet
Presence of geolocation	nominal	Existence of geolocation data in a tweet
Sensitive flag	nominal	Existence of user-set sensitive flag in the tweet's metadata

URL Features. A number of case studies on detecting phishing emails have already revealed that URL features contribute to the identification of phishing sites. Many phishing sites abuse browser redirection to bypass blacklists therefore the number of redirections between the initial URL and the final URL is another feature we collected. We have also identified extra features from WHOIS for our collected sample domains. We identified five URL features shown in Table 3.

Table 3. Identified URL Features

Feature	Type	Description
URL length	numeric	Number of characters found in a URL
Domain length	numeric	Number of characters found in the domain of a URL
Redirection count	numeric	Number of redirection hops between the initial URL and the final URL
Age of domain	numeric	Number of days since the registered date of the domain
Dot count	numeric	Number of dots (.) found in a URL. For example, www.auckland.ac.nz has 3 dots

Feature Analysis. Prior to designing our phishing detection technique, two analyses were executed to gain better insight into the feature set and eliminate any visible noise features in the early phase. In addition to the datasets we collected, we have also obtained datasets from a group of authors who have completed similar work in Miller *et al.* [13].

Table 4 presents the ranking of features based on $\overline{\chi^2}$ values from our sample set. It can be noted that the most important features are URL count, age of account and dot count. Research on Twitter spammers presented by Benevenuto et al. displays a similar behaviour [5].

Table 4. Results of χ^2 computation on the sample set

Feature	$\overline{\chi^2}$	$\overline{p-value}$	Feature	$\overline{\chi^2}$	$\overline{p-value}$
URL count	0.1092	0.8501	Listed count	19.3584	0.3060
Age of account	0.3190	0.7264	Url length	50.1056	0.0001
Dot count	0.3547	0.7114	Favourites count	1381.7993	0.1552
Redirection count	0.7762	0.4426	Following count	1526.8343	0.0392
Mention count	0.9181	0.4675	Age of domain	2558.3320	0.0404
Hashtag count	1.2330	0.3447	Follower count	3156.6372	0.0893
Domain length	10.4100	0.0101	Tweets count	26550.8693	0.0024

6 PDT Approach

In this section we describe our PDT approach in detail. In Sect. 6.1 we discuss the DBSCAN technique which was used in Phase 1, and in Sect. 6.2 we discuss the Der-TIA approach adapted for our technique.

6.1 Phase 1: DBSCAN

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a density based data clustering algorithm. The principal idea of this algorithm is that a set of points closely packed together forms a cluster and any data points in the low density region are labelled as outliers.

Cluster Classifier. DBSCAN can cluster data only and is unable to assign them into phishing or non-phishing categories. In order to carry out the assignment, Ordinary Least Squares (OLS) method was used, which computes the linear approximation for each pairwise feature. Distinctive linear regressions for the feature pairs were identified for all clusters. They were further analysed and patterns were observed as shown in Fig. 1 to draw four rules as described in Table 5. This was carried out on the initial set.

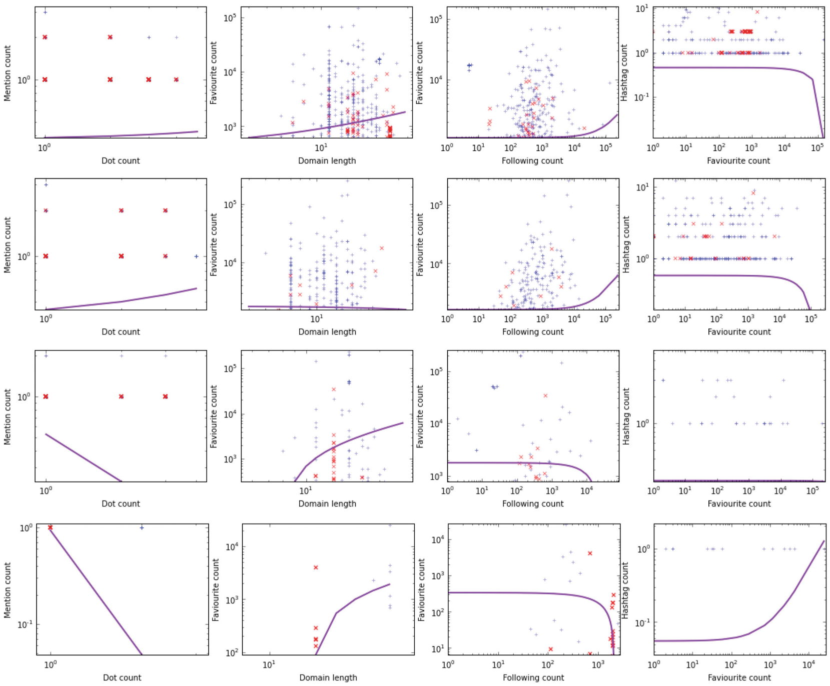

Fig. 1. Observed OLS patterns for pairwise features in clusters plotted on log-log charts. Red markers represent phishing samples and blue markers represent legitimate samples. The first two clusters are labelled as legitimate and the remaining clusters are labelled as phishing. (Color figure online)

Table 5. Cluster classification rules (β is the regression coefficient in OLS method)

Rule	Feature pair	Condition	Decision
Rule 1	Dot count vs. Mention count	$\beta < 0$	Phishing
Rule 2	Domain length vs. Favourites count	$\beta > 0$	Phishing
Rule 3	Following count vs. Favourites count	$\beta < 0$	Phishing
Rule 4	Hashtags count vs. Favourites count	$\beta > 0$	Phishing

The minimum requirement size of a cluster is 30 data points. If the size is less than 30, the cluster is considered as insufficient to comprise any one of the patterns we identified earlier. Such clusters are then indeterminate. The second phase is necessary to filter the phishing data point from the clusters that were indeterminate. Additionally, a cluster is determined by this procedure when at least 3 out of 4 labels are the same.

If the numbers of phishing and non-phishing labels are the same and the majority cannot be found from the results, then the cluster is also labelled as indeterministic as shown in Table 6. Data within clusters that were indeterminate were then pass through Phase 2.

Table 6. Cluster classification results

Cluster	Rule 1	Rule 2	Rule 3	Rule 4	Label
C1	Legitimate	Phishing	Legitimate	Legitimate	Legitimate
C2	Phishing	Phishing	Phishing	Legitimate	Phishing
C3	Legitimate	Phishing	Legitimate	Legitimate	Legitimate
C4	Phishing	Phishing	Phishing	Phishing	Phishing
C5	Legitimate	Phishing	Phishing	Legitimate	Indetermined

6.2 Phase 2: DeR-TIA

In order to classify indeterminate data points from the first phase, a technique called DeR-TIA by [19] is used. DeR-TIA combines the results of two different measures techniques to detect abnormalities in recommender systems.

DegSim. The first part of the algorithm finds the similarities between data sets using the Pearson correlation algorithm in conjunction with the k-Nearest Neighbour algorithm. The Pearson correlation algorithm yields a linear correlation (dependencies) between two data points.

$$W_{pq} = \frac{\sum_{f \in F}(V_{pf} - \overline{V_p})(V_{qf} - \overline{V_q})}{\sqrt{\sum_{f \in F}(V_{pf} - \overline{V_p})^2 (V_{qf} - \overline{V_q})^2}} \tag{1}$$

where F is the set of all features, V_{pf} is the value of feature f in the data point p and $\overline{V_p}$ is the mean value of data point p. The outcomes from the above formula range between −1 and 1. A positive value indicates that the inputs have a positive correlation and vice versa. If there is no correlation found, the outcome will be 0 and the boundary values, −1 and 1, illustrate that there are very strong correlations. The closer the outcome value is to the boundary limits, the closer the data points to the line of best fit.

The DegSim attribute is defined as the average Pearson correlation value of the k-Nearest Neighbours (k-NN) over the k.

$$DegSim = \frac{\sum_{p=1}^{k} W_{pq}}{k} \tag{2}$$

where W_{pq} is the Pearson correlation between data point p and q where k is the number of neighbours.

RDMA. The second part of the algorithm is Rating Deviation from Mean Agreement (RDMA) technique. This technique measures the deviation of agreement from other data points on the entire dataset. The RDMA measure is as follows:

$$RDMA_p = \frac{\sum_{f \in F} \frac{|V_{pf} - \overline{V_f}|}{NV_f}}{N_p} \tag{3}$$

where N_p is the number of features data set p has, F is the set of all features, V_{pf} is the value of feature f in the data point p, $\overline{V_f}$ is the mean value of feature f in the entire sample set and NV_f is the overall number of feature f in the sample set.

Once values for RDMA and DegSim are computed, the dataset of each property is split into two clusters to find the legitimate and phishing groups using the k-means algorithm where the value of k fixed at 2. The labels of clusters are determined by the cluster size. As discussed earlier, the majority of the URLs in tweets are non-phishing, therefore the larger cluster is classified as legitimate. Then the phishing clusters from RDMA and DegSim are intersected to get the final set of phishing data points.

7 Experiments

To compare the performance of our technique, we anaylsed our technique against DBSCAN. The two major reasons we chose DBSCAN as our baseline algorithm were (1) DBSCAN is an unsupervised learning technique that does not require network data features [12], and (2) DBSCAN was used in Phase 1 of our technique, thus we are able to measure improvement rate of our second phase compared to using only DBSCAN.

7.1 Experimental Setup

The experimental environment is set up with a *t2.micro* compute instance running on Amazon Elastic Compute Cloud platform. *t2.micro* instances are configured with an Intel Xeon 2.5 GHz CPU and 1 G RAM. Public tweet stream crawler and analysis tools for this research are written in Python with scientific libraries such as numpy, scipy, mathplotlib and StatsModels. 15 distinct tweet sets (VS) were sampled. Table 7 summaries collected sample sets for this research.

Table 7. Samples information

Datasets	Collection time (UTC)	#tweets collected	#tweets with URLs	#legitimate tweets
VS 1	2015-04-30 04:06:01	28685	7459	7052
VS 2	2015-04-30 06:03:01	22553	7076	6381
VS 3	2015-04-30 10:02:01	26818	7900	7375
VS 4	2015-05-01 08:22:01	44216	12059	11164
VS 5	2015-05-01 22:47:01	55078	14213	13734
VS 6	2015-05-02 00:15:01	54100	13343	12648
VS 7	2015-05-02 06:51:01	45688	12819	12369
VS 8	2015-05-02 12:14:01	62837	14830	14346
VS 9	2015-05-02 16:41:02	68233	15907	15128
VS 10	2015-05-02 18:24:01	62074	14444	13809
VS 11	2015-05-03 00:46:01	55110	13991	13629
VS 12	2015-05-03 02:18:01	60620	13570	13093
VS 13	2015-05-03 06:55:01	51584	12727	12303
VS 14	2015-05-03 08:35:01	49417	12605	12206
VS 15	2015-05-03 10:03:01	53932	13623	12917

In order to evaluate the correctness of our experiment method based on the features set collected, the commonly accepted information retrieval metrics are used. Accuracy, precision, recall and f-measure are measures used to quantify the effectiveness of PDT.

7.2 Experimental Results

By using PDT we notice an average improvement of 63 % in accuracy and an average improvement of 47 % in f-measure compared against DBSCAN. We also note that although there were no increase in precision there was a 62 % increase in recall value when we compared DBSCAN to PDT. We do note that the original precision in DBSCAN was relatively high with an average pf 99.7 %. Any further improvements would have been marginal.

We discovered that the accuracy of the results from validation samples are noticeably higher than the initial experiment results. It can be speculated that with the underlying dynamic behaviour of social media, that either Twitter

Table 8. Obtained metrics from different samples

Datasets	DBSCAN				PDT			
	Accuracy	Precision	Recall	f-measure	Accuracy	Precision	Recall	f-measure
VS 1	0.4497	0.9975	0.4454	0.6158	0.9858	0.9989	0.9861	0.9924
VS 2	0.4624	0.9936	0.4595	0.6284	0.9816	0.9968	0.9828	0.9897
VS 3	0.3147	0.9920	0.3371	0.5032	0.9542	0.9949	0.9558	0.9750
VS 4	0.2212	0.9996	0.2247	0.3670	0.9671	0.9923	0.9720	0.9820
VS 5	0.2797	0.9982	0.2799	0.4372	0.9780	0.9971	0.9800	0.9885
VS 6	0.2674	0.9983	0.2717	0.4271	0.9684	0.9955	0.9711	0.9831
VS 7	0.3842	1.0000	0.3889	0.5600	0.9874	0.9998	0.9871	0.9934
VS 8	0.3481	1.0000	0.3518	0.5205	0.9873	0.9994	0.9874	0.9934
VS 9	0.3672	0.9986	0.3788	0.5493	0.9894	0.9991	0.9898	0.9944
VS 10	0.3853	0.9967	0.3948	0.5656	0.9851	0.9979	0.9865	0.9922
VS 11	0.3731	0.9969	0.3744	0.5444	0.9873	0.9987	0.9883	0.9935
VS 12	0.3943	0.9990	0.3975	0.5688	0.9892	0.9996	0.9892	0.9944
VS 13	0.3570	0.9991	0.3601	0.5294	0.9899	0.9996	0.9900	0.9948
VS 14	0.3397	0.9972	0.3501	0.5192	0.9793	0.9988	0.9798	0.9892
VS 15	0.3243	0.9991	0.3336	0.5002	0.9928	0.9994	0.9930	0.9962
StdDev	0.0645	0.0023	0.0626	0.0697	0.0107	0.0022	0.0098	0.0059

user's usages or phishing trend or both have been changed over the 208 day period since the time of the initial sample collection. We postulate that this is due to the adaptive nature of our unsupervised learning based technique, which can adapt to the continuously-changing Twitter ecosystem. The samples we collected over the three days show consistency in all metrics as supported by low standard deviation values. The Mann-Whitney test on the f-measures supports that PDT is significantly more accurate than DBSCAN with input parameters as $U = 16$, Z-ratio $= -4.2038$, $P \leq 0.05$ two-tailed (Table 8).

8 Conclusions

In this research we proposed a technique, called PDT, which is a two phase approach adapting both DBSCAN and DeR-TIA to improve the coverage. Our technique PDT shows promising outcomes. We showed that our technique produced high accuracy, precision, recall, and f-measure values compared to previous technique namely DBSCAN. Moreover PDT can adapt over time for phishing detection with behavioral changes in Twitter. We have shown that our technique worked over data collected from different periods of time. Using other traditional supervised methods we would not have been able to guarantee that it could adapt to the changing nature of the Twitter data.

For further development, phishing detection through a sliding window over the stream of new tweets would be useful. A fixed size window accepts new data from Twitter streams and eliminates obsolete data from the window pool. Thus ensuring recency of information evaluated and may increase the accuracy of the results produced.

References

1. Phishtank — join the fight against phishing, June 2015. https://phishtank.com
2. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit, pp. 60–69. ACM (2007)
3. Aggarwal, A., Rajadesingan, A., Kumaraguru, P.: Phishari: automatic realtime phishing detection on twitter. In: eCrime Researchers Summit (eCrime), 2012, pp. 1–12. IEEE (2012)
4. Amleshwaram, A.A., Reddy, N., Yadav, S., Gu, G., Yang, C.: Cats: characterizing automation of Twitter spammers. In: 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), pp. 1–10. IEEE (2013)
5. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on Twitter. In: Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS), vol. 6, p. 12 (2010)
6. 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. ACM (2011)
7. Dunlop, M., Groat, S., Shelly, D.: Goldphish: using images for content-based phishing analysis. In: 2010 Fifth International Conference on Internet Monitoring and Protection (ICIMP), pp. 123–128. IEEE (2010)
8. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide Web, pp. 649–656. ACM (2007)
9. Garera, S., Provos, N., Chew, M., Rubin, A.D.: A framework for detection and measurement of phishing attacks. In: Proceedings of the 2007 ACM Workshop on Recurring Malcode, pp. 1–8. ACM (2007)
10. Klien, F., Strohmaier, M.: Short links under attack: geographical analysis of spam in a url shortener network. In: Proceedings of the 23rd ACM Conference on Hypertext and Social Media, pp. 83–88. ACM (2012)
11. Lee, S., Kim, J.: Warningbird: a near real-time detection system for suspicious urls in Twitter stream. IEEE Trans. Dependable Secure Comput. **3**, 183–195 (2013)
12. Liu, G., Qiu, B., Wenyin, L.: Automatic detection of phishing target from phishing webpage. In: 20th International Conference on Pattern Recognition (ICPR), pp. 4153–4156, August 2010
13. Miller, Z., Dickinson, B., Deitrick, W., Hu, W., Wang, A.H.: Twitter spammer detection using data stream clustering. Inf. Sci. **260**, 64–73 (2014)
14. Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., Zhang, C.: An empirical analysis of phishing blacklists. In: Sixth Conference on Email and Anti-Spam (CEAS), California, USA (2009)
15. Wang, A.H.: Don't follow me: spam detection in Twitter. In: Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT), pp. 1–10. IEEE (2010)

16. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: Cantina+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. (TISSEC) **14**(2), 21 (2011)
17. Yardi, S., Romero, D., Schoenebeck, G.: Detecting spam in a Twitter network. First Mon. **15**(1) (2010)
18. Zhang, H., Liu, G., Chow, T.W., Liu, W.: Textual and visual content-based anti-phishing: a Bayesian approach. IEEE Trans. Neural Netw. **22**(10), 1532–1546 (2011)
19. Zhou, W., Wen, J., Koh, Y.S., Alam, S., Dobbie, G.: Attack detection in recommender systems based on target item analysis. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 332–339. IEEE (2014)