Information Sciences 289 (2014) 133–147

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Feature set identification for detecting suspicious URLs using Bayesian classification in social networks

Chia-Mei Chen *, D.J. Guan, Qun-Kai Su

National Sun Yat-sen University, Kaohsiung, Taiwan, ROC

ARTICLE INFO

Article history:
Received 30 August 2013
Received in revised form 12 June 2014
Accepted 24 July 2014
Available online 9 August 2014

Keywords:
Social network
Anomaly detection
Bayesian classification

ABSTRACT

Social network services (SNSs) are increasing popular. Communicating with friends forms a social network that can be used to promptly share information with friends. In targeted attacks, SNSs are often used to collect personal information and craft attacks based on a specific user profile. Malware can be used to facilitate social relationship, sends messages containing malicious URLs, lures users to click on these URLs by employing social engineering techniques; then replicates through the social network over and over again. Because users are curious and trust in their friends, they typically click on malicious URLs without verification. In this study, a feature set is presented that combines the features of traditional heuristics and social networking. Furthermore, a suspicious URL identification system for use in social network environments is proposed based on Bayesian classification. The experimental results indicate that the proposed approach achieves a high detection rate.

1. Introduction

Based on advances in information technology, websites offer various convenient web services such as information retrieval, chat rooms, Web 2.0-based services, blogs, albums, and multimedia sharing. Social network services (SNSs), such as Facebook, Twitter, and MySpace, have recently proliferated offering interactive information platforms that allow users to share and to interact. Based on Facebook statistics [6], Facebook owns 845 active members monthly. A TrendMicro report [19] indicated that the percentage of employees using SNS increased from 19% in 2008 to 24% in 2010. In addition, based on a survey Pew Research [17], the trend toward using SNSs has annually increased among all age groups; thus, SNSs are increasingly popular and essential.

Incident investigation reports have indicated that cybercrimes, such as targeted attacks or advanced persistent threats (APTs) often use SNSs to collect personal information and launch social engineering attacks [25,29,12]. In other words, the convenience of SNSs facilitates potential cyber attacks on SNS platforms. For example, a social-network-based worm spreads by attempting to steal account information and infect additional users by using a social engineering trick that sends malicious URL posts or emails. Because SNS users typically trust their friends, they are often breached by these worms, which rapidly spread through the friendship connections of victims.

Malware programs often leverage short URL and blog services are often used by malware to disguise original URLs and evade security inspections, such as blacklist filtering applications. URL shortening transfers original URLs through shortened URLs by using redirection. Because URL shortening is often abused, these providers may find themselves blacklisted.

* Corresponding author.

http://dx.doi.org/10.1016/j.ins.2014.07.030
0020-0255/

SNSs comprise wide range of interactive and sharing services is offered by SNS. Walls in Facebook which combine bulletin and message boards, enable board members post messages. In instant message environments, only those in a contact list can send messages, whereas in SNSs, a cluster of friends formed through friendship links or a cluster of members who share common interests are not limited to contact lists, and can post messages that include malicious URLs.

Walls are used on personal, club, and fan pages; access to personal and club pages is protected using access controls, whereas fan pages cannot. All users can access fan pages without requesting permission from the page owner. A Websense survey [30] indicated that 10% of URLs posted to in Facebook were malicious links; thus, users who access popular fan pages may risk security breaches. Hackers can spread attacks by simply posting messages containing malicious links on the most popular fan walls; multiple fans are likely to click on such links.

Compared with spam, posts containing malicious URLs are faster and more effective. As long as the content of the post addresses hot topics, it can catch the attention of numerous users. In this study, a set of heuristic features and Bayesian classification are proposed for detecting malicious URLs in SNSs. According to the findings, malicious web links in a post exhibit domain and social anomalies that differ from those of typical links. The proposed detection method involves using a naive Bayesian model to detect social network posts that contain malicious URLs based on anomalies in the URL domain and unusual posting behaviors.

2. Related work

This section introduces spam, phishing, and methods of detecting suspicious URLs in online messages and social networks. Classification algorithms are introduced along with their related network security applications in detecting anomalies.

Zhang et al. [32] proposed a content-based method for detecting phishing web sites, suggesting that phishing sites are created based on minor modifications from the authenticated sites and exhibit low page ranks in the Google search results. A set of heuristics was proposed based on domain name, lexical signatures of web links, and the HTML content of web pages. Five keywords were extracted from each web page based on TF-IDF (term frequency/inverse document frequency) algorithm and the Google search was applied to verify the website authenticity.

According to McGrath et al. [18], a brand name should appear in the URL of a web site. They collected and analyzed the URLs of phishing and non-phishing websites, determining that diverse countries host phishing sites, phishing domains are rarely hosted in their registered country, and phishing domains last approximately 3 days.

Fette et al. [8] extracted email features such as HTML tags, the number of links, use of javascript, and number of domains, to distinguish phishing emails by using a support vector machine (SVM). Bergholz et al. [3] proposed using email features namely, the structure of the email body, web link properties, and a keyword list, which were generated using dynamic Markov chain training and class-topic models. Their results indicated that using these features improved the detection rate. Abu-Nimeh et al. [1] chose the 43 most popular keywords as features and evaluated the performance level by using various machine learning classification algorithms.

Ma et al. [14] adopted semantic features from McGrath [18] and bag-of-words features from Kolari et al. [13]. In addition, Ma et al. addressed features specific to the hosted machine such as the IP address, WHOIS information, domain name, and geographic location. Machine learning algorithms namely, the native Bayesian theorem, Support Vector Machine (SVM), and logistic regression, were applied to evaluate the detection of suspicious URLs when using various combinations of features and data sets. Their subsequent studies [15,16] have yielded the similar conclusions, indicating that the features of URL semantics and host information are essential for identifying malicious web links when a suitable machine learning technique is applied.

Online messaging is a real-time communication service. Morse et al. [21] noted that online messaging networks are scale-free, and the distribution of network node connectivity follows power law. In other words, highly connective nodes are likely to be connected with other nodes. Therefore, worms that infect highly connected nodes can rapidly propagate throughout a network, making it difficult to completely remove such worms. Thus, identifying methods of effectively detecting or preventing online messaging worms is critical.

Guan et al. [10] proposed a suspicious URL detection method used in instant message environments, noting the discrepancies between human and robot interactions and proposing features based on anomalous user behaviors, such as response time entropy, delay time entropy, and the domains of attached links (e.g. domain rank, lexical features, and WHOIS information). The experimental results demonstrated that the features were effective and the scoring model attained a sufficient performance level. Guan el at. demonstrated that distinct interaction timing features are distinguishable in online messages. Although the proposed features might not be applicable for detecting malicious URLs in social networks, the unique social communication patterns of a user might be essential for detecting malicious activities.

Social networks exhibit scale-free and small-world network characteristics [20]. Small-world networks consist of multiple clusters that exhibit small average shortest path lengths and high clustering coefficients. Therefore, the nodes within a cluster are densely connected. A cluster in a social network may be a friend group or a club of members sharing common interest. In these networks worm infection may propagate through friend groups and clubs, spreading more rapidly compared with infections in instant message environments.

According to Xu et al. [31], social network worm attacks behave similarly to internet worm attacks; however, the methods of detecting internet worms do not apply to social networks. Detecting internet worms often involve observing anomalous behavior in network traffic or the infected host; by contrast, each client host and the required traffic or host behaviors cannot be monitored on social network sites. In addition, infected clients might behave similarly to uninfected users, posting messages, updating personal account information, or joining new groups. Therefore, detecting social network attacks is extremely challenging.

The first and most influential social network worm, Koobface [12,28], attacked Facebook and MySpace in December 2008, primarily attacking by sending messages that contained links to malicious websites and leveraging various SNSs information sharing, emails, and social network applications, as vehicle of spreading malware [27,2]. Koobface could rapidly propagate and was difficult to remove.

Thomas et al. [27] observed the infection and detection rate of Koobface for one month. Compromised social network accounts distributed malicious links to more than 213 thousand users. Blacklist services required 4 days to react and only identified 27% of malicious URLs. During this period, 81% of users clicked on Koobface spam. Efficient detection methods are required to react the dynamic changing of suspicious URLs and to prevent further infection.

Xu et al. [31] used a deployed surveillance network over a social network to collect and analyze worm propagation behavior. Decoy accounts were constructed to connect with popular users who were highly connected. The existence of a worm was identified based on the similarity of messages collected from the decoy accounts. However, regular posts, such as breaking news stories, might be sent in broadcasts that exhibit similar patterns as worm propagation. Detection methods based on only message propagation behavior and ignoring message content might cause false alarm. In addition, deploying surveillance might not be applicable to real environments. Practical problems, such as how to choose popular users, the number of decoy accounts, and willingness to have decoy accounts as friends, should be solved before applying the proposed social worm detection to real networks.

Stringhini et al. [26] applied a decoy profile approach for collecting social information from users. Social features were used to identify spammers in social networks without considering the characteristics of malicious web links or content in posts. It was assumed that a spammer would send numerous friend requests but was not a real-life friend of others, whereas hackers might steal user accounts to issue malicious posts. Considering web links or content could enhance the detection rate.

Jin et al. [11] developed a spam detection system for use in social networks, proposing three types of features: image content, text, and social network features. The social network features address the individual characteristics of user profiles and their behaviors. It was assumed that normal users and spammers exhibited distinct posting behaviors; however, Jin et al. did not clearly describe clearly what the features are. In addition, users might provide limited information to the public, making it impossible to extract the required anomaly characteristics.

Based on the relevant literature, the characteristics of malicious web links are crucial when classifying suspicious URLs because links formerly sent through email are now being sent through social networks. A post that contains a suspicious URL is similar to an email that contains a phishing link. Spammers or attackers leverage social trust to propagate malicious posts in social networks. Therefore, the features of malicious web links and social relationship should be considered when attempting to detect suspicious web links.

Previous studies on detecting malicious activities in online messages [10] and social networks [11,26] have suggested that the unique social behaviors of users are essential.

Various attack tactics are used to compromise the security of a host or network, and all attacks generate traffic anomalies. After observing hidden non-stationary traffic patterns in anomalous traffic, Palmieri and Fiore [22] proposed a network anomaly classification method applying recurrence quantification analysis; they concluded that combining the proposed detection approach with SVM-based machine learning yielded a reliable system. Fiore et al. [9] constructed a self-learning model that could characterize current behaviors based on past events. The results demonstrated that efficient detection relies on the accuracy of the self-learning model.

Machine learning approaches are commonly used to classify malicious URLs and anomalies. The subsequent paragraphs introduce logistic regression, SVM, and Bayesian classification.

Logistic regression is used to predict the outcome of a binary dependent variable based on various predictor variables; this involves a generalized linear model or binomial regression as follows:

$$\log \frac{P(x;\beta)}{1-P(x;\beta)} = \beta^T x. \tag{1}$$

where $x = (x_1, x_2, \ldots, x_p)$ is the vector of prediction model p, y is a binary response variable, and β is a vector $p \times 1$ of the regression parameter.

Based on the properties of multivariate statistics, logistic regression can be used to observe and analyze multiple inputs and predict an outcome. It has been applied in diverse areas such as biostatistics, sociology, and marking. Applying logistic regression to binary classification problems also yields strong performance level. However, logistic regression requires making statistical assumptions that might not be practical in certain situations and sensitive to missing data.

SVM represents training data as points in space and locates one hyperplane in order to classify the data into categories. The SVM algorithm is based on statistical learning theory. Fig. 1 shows a classification scenario using an SVM. The points

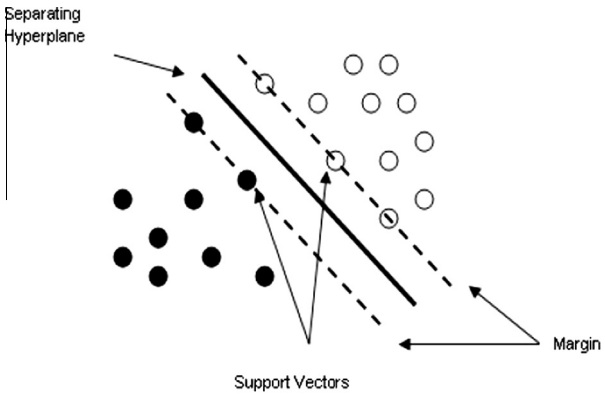

Fig. 1. An illustration of SVM [1].

representing training data are support vectors and the solid line represents the separating hyperplane used for classifying the test data. Training can consume substantial computational resources. Noisy data could cause overfitting and degrade the detection performance [1].

Bayesian classification is based on probabilistic model specification. It employs naive Bayes assumptions: features that describe data instances are conditionally independent given the classification hypothesis. The maximum a posteriori estimation is computed by incorporating prior information. It is considered a generative approach to classification and has been successfully applied in medical diagnosis, text classification, and spam detection.

In addition, Bayesian classification is insensitive to noisy data [23] and incremental improvement properties enhance the level of performance when the volume of data increases. These properties are suitable for use in virtual social environments. Therefore, Bayesian classification was used to establish the classification process in the proposed system.

3. Proposed approach

This study involved detecting malicious URLs in social network environments. Based on relevant literature and incident analysis [25,29,12,28], attackers may use SNSs as vehicles, using compromised user accounts to post messages that contain malicious URLs. Social network users typically trust the information that their friends submit in posts and feeds; thus, they become the victims of social engineering attacks. Therefore, in addition to the traditional attributes of malicious URLs, social networking heuristics should be addressed to facilitate identifying malicious URLs in social networks. Certain malicious URLs are disguised using blogs or URL shortening service; this increases the difficulty of detecting malicious posts and feeds. Malicious URLs used in social networks may make use of the trust and social relationships which resembles phishing websites in spam; thus, multiple sets of features are proposed for detecting spam or malicious URLs. In this study, Facebook was used as the social network environment and posts were collected using Facebook API.

Fig. 2 shows the proposed system and overall process for detecting SNS-based malicious URLs. In the first module, data collection, posts are collected including time and content. Posts that lack URL information are considered benign. In the second module, feature extraction, the proposed features (elaborated in subsequent sections) are retrieved and a feature vector is constructed for classification. In the third module, the Bayesian classification model,posts are classified based on a pre-trained classification model.

3.1. Feature selection

(1) F_1**: Dash Count in Hostname**:
Zhang et al. [32] and Lin [10] have supported the importance of lexical features in detecting malicious URLs. A preliminary analysis on the blacklists indicated that numerous malicious URLs contain dashes, whereas legitimate URLs rarely include dashes. Therefore, the number of dashes in the host name was used as a lexical feature in this study.

(2) F_2**: Longest Domain Label**:
This can be assumed. Legitimate websites typically use meaningful, short, and easy-to-remember terms as domain names; compared with the domain names of benign websites, those of malicious websites are typically longer, and may not combine meaningful terms. Therefore, this feature extracting a term that represents the meaning of the website. For example, the longest domain label of "www.facebook.com" is "facebook," which has a feature value of eight; thus, the length is "facebook" is eight. The longest value is used to compute the feature value when multiple URLs are listed in a post.

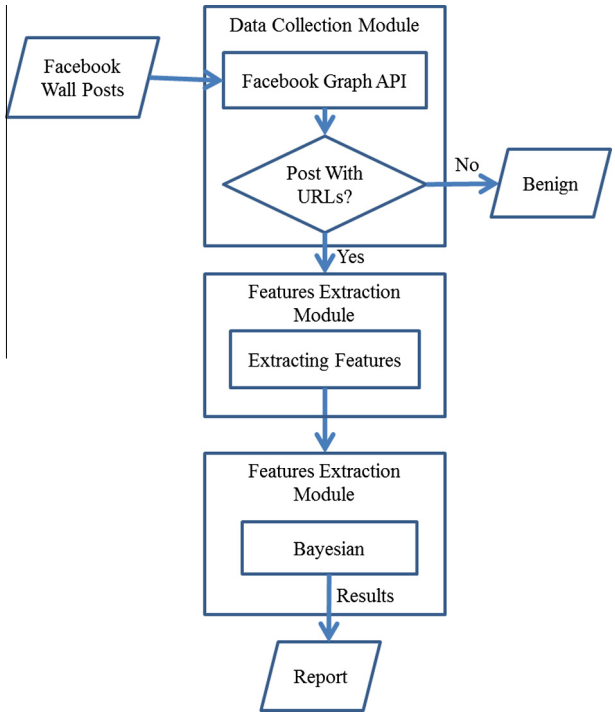

Fig. 2. The proposed system architecture.

(3) F_3**: Domain Rank**:
Guan's study [10] indicated that the Google search reputation or rank of a website yields strong classification results. Because the API returns a limited number of search results, a website was considered normal if it ranked within the first four search results; the lowest rank was used when multiple URLs were listed in a post.

(4) F_4**: Domain Age**:
A normal domain typically exhibits a long history and extended domain registration. Relevant studies have suggested that malicious URLs exhibit domains registered at a future date, or lack registration dates. Most malicious domains are promptly taken down; thus, the domain age feature is considered.

(5) F_5**: URL count**:
Attackers may post multiple URLs in a post, targeting the diversified vulnerabilities or interests of users. Attackers seek to flood a wall, whereas normal users do not typically compose posts that contain multiple URLs. Fette et al. [8] used a similar feature for detecting phishing; thus, the URL count is used a feature in the proposed detection method.

(6) F_6**: Similar Message Count from a User**:
Xu et al. [31] indicated that news feeds posted by worm are automatically generated. A virus may present a limited number of meaningful sentences, and it is highly possible that malicious messages involve similar content. In contrast to compromised accounts or viruses, a normal user rarely posts similar content several times. Therefore, this feature represents anomalous behavior. A fuzzy string comparison was used to compute the similarity of message content. Similarity is defined as the length of the longest common subsequence (LCS) of two feeds compared with the average of their string lengths, as follows:

$$\text{Similarity}(str1, str2) = 2 \times \frac{|\text{LCS}_{str1,str2}|}{|str1| + |str2|}. \tag{2}$$

(7) F_7**: Similar Message Count from Different Users**:
Attackers may use multiple compromised accounts to post suspicious posts and highly connected users might receive the same or similar information multiple times. Therefore, this feature is used to count the number of similar messages that distinct user accounts post on a wall.

3.2. Bayesian classification model

Two labels were defined in sample space: benign (B) and malicious (M). The distribution probability can be expressed follows:

$$\begin{cases} P(B) = \frac{|B|}{|B|+|M|}; \\ P(M) = \frac{|M|}{|B|+|M|}. \end{cases} \tag{3}$$

The prior probability of each feature vector (F, f) is indicated as follows:

$$\begin{cases} P(F_i = f_i|B) = \frac{P((F_i=f_i)\cap B)}{P(B)}, & \text{for } i = 1,2,\ldots,n, \quad n \in N; \\ P(F_i = f_i|M) = \frac{P((F_i=f_i)\cap M)}{P(M)}, & \text{for } i = 1,2,\ldots,n, \quad n \in N. \end{cases} \tag{4}$$

The posterior probability of the two labels can be computed as follows after calculating prior probabilities by using Formula 4. This equation is as follows:

$$\begin{cases} P(B|F_1 = f_1,\ldots,F_n = f_n) = \frac{P(B)\prod_{i=1}^{n} P(F_i=f_i|B)}{P(F_1,\ldots,F_n)}, & n \in N; \\ P(M|F_1 = f_1,\ldots,F_n = f_n) = \frac{P(M)\prod_{i=1}^{n} P(F_i=f_i|M)}{P(F_1,\ldots,F_n)}, & n \in N. \end{cases} \tag{5}$$

If the observed features indicate a strong chance that a post is a malicious post, it is considered a malicious post. Hence, the classification rule is defined as follows:

$$P(B|F_1 = f_1,\ldots,F_n = f_n) \leqslant P(M|F_1 = f_1,\ldots,F_n = f_n). \tag{6}$$

4. Experimental evaluation

4.1. Data collection

Thomas et al. [27] indicated that social-network-based worms randomly select and participate in popular clubs. Information regarding the most popular walls or pages (based on the number of fans) is typically publicly available on social network websites [4,7,5]. Because popular clubs involve numerous fans, worms and social engineering attacks can spread widely and rapidly. Therefore, popular walls and clubs are at high risk of being attacked. Furthermore, attackers typically leverage hot topics related to walls and applies social engineering tricks to launch malicious posts or feeds.

Based on fan page rankings [4,7,5], test data were collected from the most popular fan pages in various geographical areas (the world, Asia, and Europe), from July 2010 to February 2011. The web links were verified using VirusTotal. The findings indicated that 5637 malicious samples and 43,473 benign samples were collected from the world, 293 malicious and 3768 benign samples were collected from Asia, and 602 malicious and 9261 benign were collected from Europe, as shown in Table 1. The experiments were performed on all the data sets.

The detection system classifies malicious as positive and benign negative. The classification evaluations are listed in Table 2 and detailed as follow:

(1) **True Positive (TP):** Malicious samples are labeled as malicious.
(2) **False Negative (FN):** Malicious samples are labeled as benign; FN is also known as Type II Error.
(3) **False Positive (FP):** Benign samples are labeled as malicious; FP is also known as Type I Error.
(4) **True Negative (TN):** Benign samples are labeled as benign.

Table 1
Data sets collected.

Top 20 from the world			Top 20 from Asia			Top 20 from Europe		
Fan pages	Malicious	Benign	Fan pages	Malicious	Benign	Fan pages	Malicious	Benign
Total	5637	43473	Total	293	3768	Total	602	9261

Table 2
Classification evaluation.

	Classified as positive	Classified as negative
Malicious samples	True Positive (TP)	False Negative (FN)
Benign samples	False Positive (FP)	True Negative (TN)

The following performance measurements are commonly used when evaluating detection: the true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), accuracy (ACC), and error rate (ERR). The measurements can be computed using the formulae listed in Table 3.

4.2. Performance evaluation

The performance evaluation comprises two parts. The first part involves assessing the proposed system by comparing it against previous research and the second involves examining the efficiency of the proposed feature sets. Test data exhibiting various ratios of malicious to benign data were evaluated to determine the performance of the proposed method in various social network environments. Various numbers of rounds were conducted, yielding nonsignificant differences in the results (<1%). The experimental results are primarily based on testing 200 rounds.

To classify malicious content, Robertson et al. [24] combined traditional security heuristics and social network information. The detection method used diverse social network information provided by Facebook, Twitter, or Google. A scanner was developed to search the friend list of a user and collect information from all posts and new feeds originating from a given user. The experimental results were based on 400 links (malicious to benign ratio = 5:5) and yields an accuracy rate of 62.5%; however, the web link data set was limited and this accuracy rate could be improved.

Stringhini et al. [26] created 900 decoy profiles on three social networks and evaluated the collected information. Four features were proposed for identifying malicious posts sent by spammers: (1) FF ratio (R): the number of friend requests that a user sent compared with the number of friends that he or she already has; (2) URL ratio (U): the number of messages containing URLs compared with the total number of messages sent by a given user; (3) message similarity (S): the number of common words in two messages compared with the average length of messages posted by the user; and (4) friend choice (F): the number of friends compared with the number of distinct first names in a friend list. Features R and F require user profile information; however, not all users disclose this information to the public. Therefore, these might not be valid for analysis in the performance comparison. It was assumed that a spammer sending similar messages would attain a high S value; however, according to the experimental results (Fig. 3), both benign and malicious users exhibit similar messages distribution. Thus, feature S might not be a distinguishable feature for classifying malicious web links.

In response to the increasing popularity of social media sharing, Jin et al. [11] proposed an online social media spam detection method based on general activity detection (GAD) clustering. The evaluated media information includes images; thus, the proposed method extracted the features of image content. Because the detection method proposed in the current study did not consider images, only text features were assessed in the performance comparison. In this evaluation, JIN refers to the text features proposed by Jin et al. and CGS refers to the proposed solution. Two classification algorithms, Bayesian and SVM, were evaluated and JIN performs similarly in both cases (Table 4). The results from Bayesian were chosen to compare with the proposed solution CGS, because the proposed solution is based on Bayesian classification as well. The proposed

Table 3
Performance measurements.

$$TPR = \frac{\text{The number of labeled as malicious in all malicious samples}}{\text{the number of malicious samples}} = \frac{|TP|}{|TP| + |FN|}$$

$$TNR = \frac{\text{The number of labeled as benign in all benign samples}}{\text{the number of benign samples}} = \frac{|TN|}{|TN| + |FP|}$$

$$FPR = \frac{\text{The number of labeled as malicious in all benign samples}}{\text{the number of benign samples}} = \frac{|FP|}{|TN| + |FP|}$$

$$FNR = \frac{\text{The number of labeled as benign in all malicious samples}}{\text{the number of malicious samples}} = \frac{|FN|}{|TP| + |FN|}$$

$$ACC = \frac{\text{correct classification in all samples}}{\text{the number of all samples}} = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|}$$

$$ERR = \frac{\text{erroneous classification in all samples}}{\text{the number of all samples}} = \frac{|FP| + |FN|}{|TP| + |TN| + |FP| + |FN|}$$

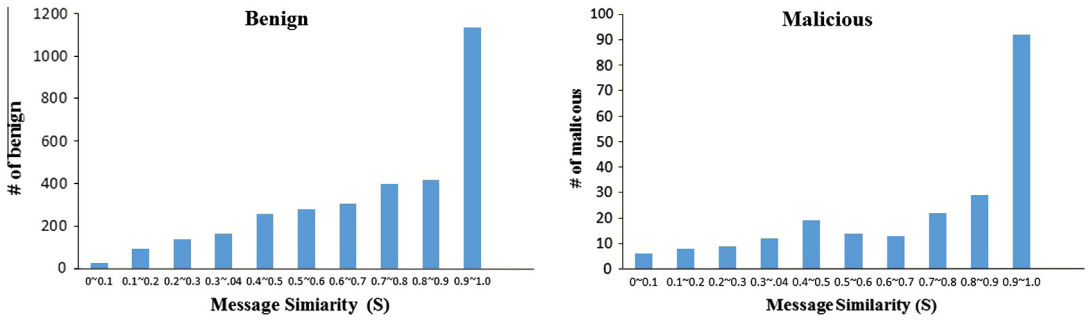

Fig. 3. The distribution of feature message similarity (S).

Table 4
Performance results of method JIN.

	5:5		1:9	
	Bayesian	SVM	Bayesian	SVM
TPR	0.74	0.73	0.30	0.23
TNR	0.88	0.89	0.96	0.80
FPR	0.12	0.11	0.04	0.02
FNR	0.26	0.27	0.70	0.77
ACC	0.81	0.81	0.90	0.90
ERR	0.19	0.19	0.10	0.10

features in CGS can be categorized into two types, domain (F_1–F_4) and social anomalies (F_5–F_7). Hence, two variations of CGS were compared with JIN to determine the performance level of the proposed feature sets, where CGS-1 and CGS-2 refer to the domain anomaly features and all features, respectively.

Fig. 4 shows the comparison results, demonstrating that the proposed features yielded superior levels of performance for both the domain anomaly feature set and all features. JIN exhibited a high FNR in both ratios (5:5 and 1:9) and its performance level could degrade in real environments in which the ratio is close to 1:9. The problem with a high FNR is that misclassifying malicious web links may cause user machines to become infected or compromised. Therefore, attaining a low FNR is essential in this classification problem. The CGS was stable and demonstrated lower FNR and higher TPR in both ratios, outperforming JIN overall.

The second part of the evaluation involved inspecting the efficiency of the proposed features; four experiments were conducted to examine the importance of the social and domain anomaly features. Experiment 1 was conducted to examine the efficiency of the domain anomaly features, F_1–F_4. Experiment 2 included the social anomaly features (F_5–F_7), and was conducted to analyze the level of detection performance when using both anomaly feature sets. In Experiment 3, short URLs were excluded from the test data to compare the detection performance level with and without including short URLs. Experiment 4 was conducted for evaluating the sensitivity of the proposed solution based on social data from various geographic areas.

A threshold for message similarity (Sim) must be determined for the F_6 and F_7 features. Various threshold values were evaluated to determine a best fit for these features. Experiment 2 evaluated the sensitivity of the similarity threshold value (Sim) based on various simulated social network environments (by using the ratios of malicious to benign).

According to Thomas et al. [27], social-network-based worms are frequently disguised using short in order to evade detection. A short URL service maps a long URL into a short form. Because of the diversity and dynamics of short URL services, short URLs might not be able to remap back to their original URLs. Twenty popular short services, such as bit.ly, goo.gl, tinyurl.com, and ow.ly, are applied to retrieve the original form of a short URL. Therefore, Experiment 3 evaluated the sensitivity of the proposed method to short URLs, comparing the performance level when including or excluding short URLs.

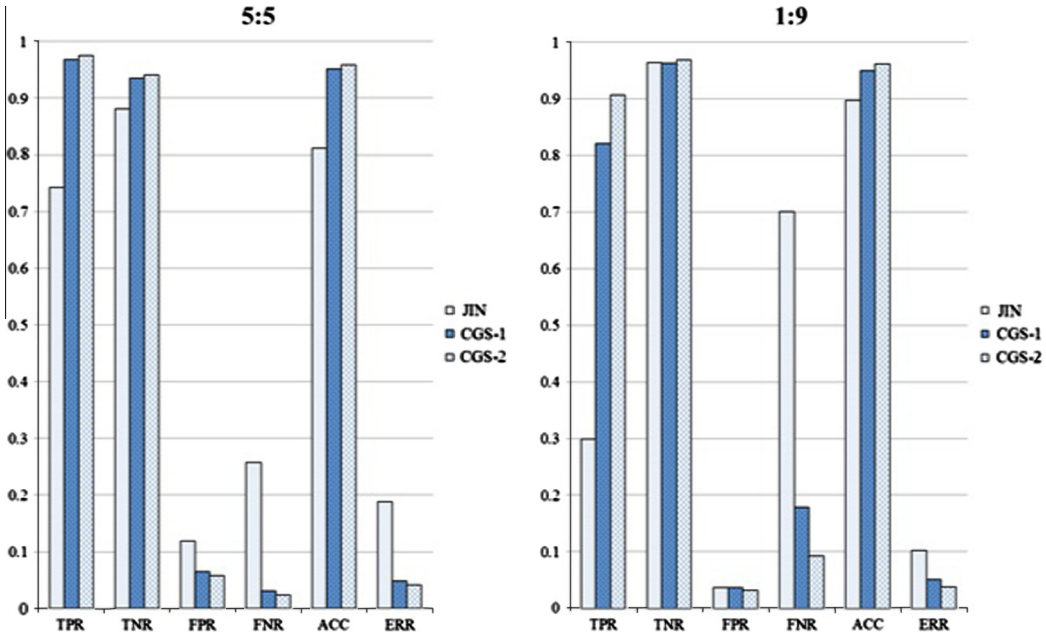

Fig. 4. The performance results of JIN and CGS.

Data were collected from various time periods and geographic areas. Experiment 4 was conducted to examine whether the proposed model could achieve stable detection performance when these factors varied.

4.2.1. Experiment 1

Combinations of the domain anomaly features from F_1 to F_4 were evaluated to determine the efficiency of the feature set. Individual features were evaluated using test data that exhibited varying ratios of malicious to benign URLs. The graphic results are shown in Figs. 5 and the numeric results are stated in Table 5. The domain anomaly feature set can be divided into distinct types: lexical features (F_1 and F_2), reputation feature (F_3), and host feature (F_4). Various combinations of features were evaluated and the results are shown in Fig. 6 and Table 5.

The results of each individual domain anomaly feature shown in Fig. 5 indicate that distinct feature types (lexical, reputation, and host) make differing contributions to detection. Features F_1 and F_3 exhibited a stable level of TNR performance

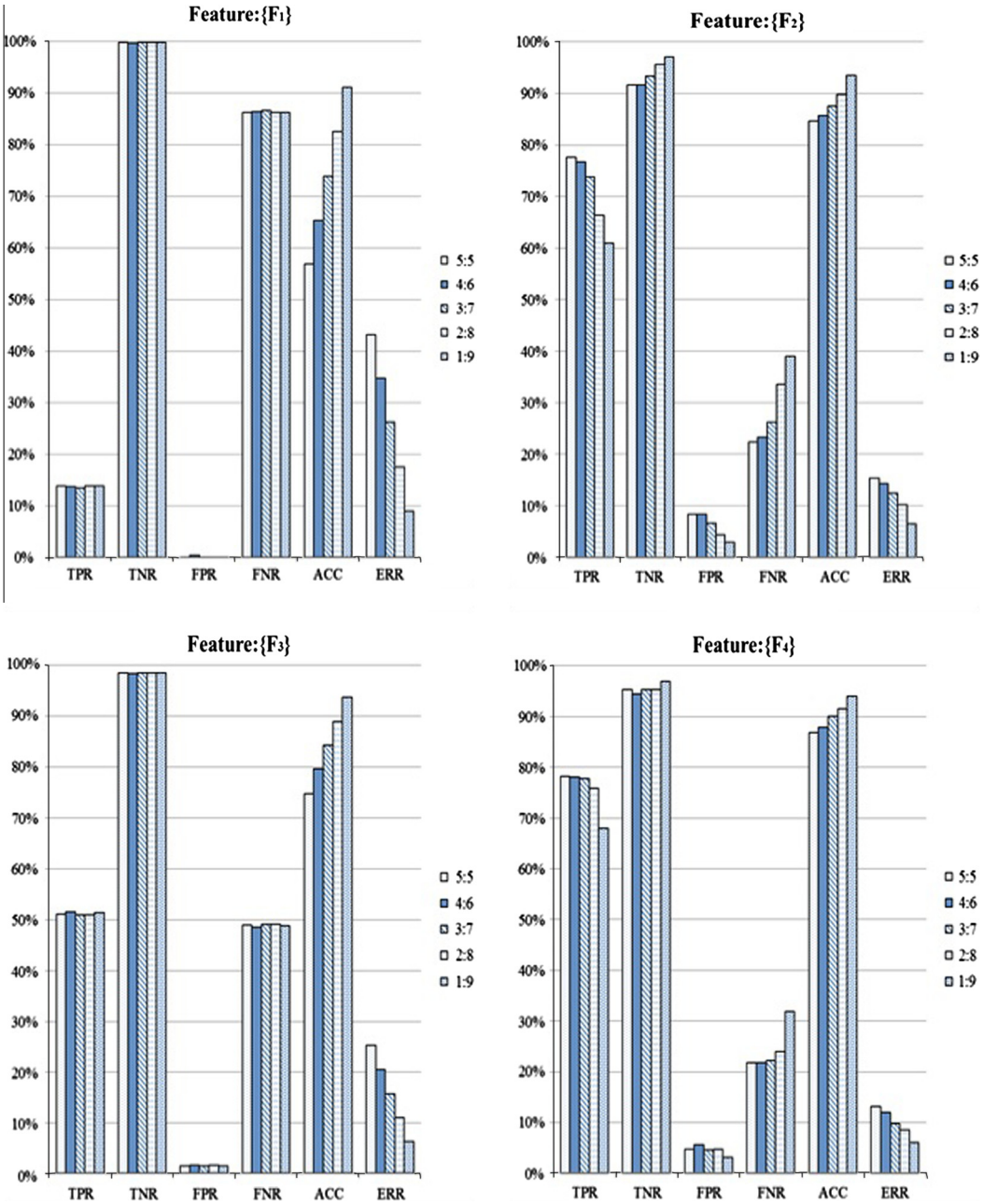

Fig. 5. The results of Experiment 1 with single domain anomaly feature.

Table 5
Numeric results of Experiment 1.

M:B (%)	TPR (%)	TNR (%)	FPR (%)	FNR (%)	ACC (%)	ERR (%)
Feature: F_1						
5:5	13.79	99.80	0.20	86.21	56.79	43.21
4:6	13.62	99.65	0.35	86.38	65.24	34.76
3:7	13.41	99.75	0.25	86.59	73.85	26.15
2:8	13.76	99.75	0.25	86.24	82.55	17.45
1:9	13.76	99.70	0.30	86.24	91.10	8.90
Feature: F_2						
5:5	77.64	91.70	8.30	22.36	84.67	15.33
4:6	76.66	91.66	8.34	23.34	85.66	14.34
3:7	73.76	93.40	6.60	26.24	87.51	12.49
2:8	66.45	95.61	4.39	33.55	89.78	10.22
1:9	60.96	97.10	2.90	39.04	93.48	6.52
Feature: F_3						
5:5	51.06	98.51	1.49	48.94	74.78	25.22
4:6	51.52	98.36	1.64	48.48	79.62	20.38
3:7	50.96	98.60	1.40	49.04	84.31	15.69
2:8	50.91	98.49	1.51	49.09	88.97	11.03
1:9	51.30	98.51	1.49	48.70	93.79	6.21
Feature: F_4						
5:5	78.26	95.31	4.69	21.74	86.79	13.21
4:6	78.14	94.46	5.54	21.86	87.94	12.06
3:7	77.80	95.39	4.61	22.20	90.11	9.89
2:8	75.94	95.36	4.64	24.06	91.47	8.53
1:9	68.06	96.89	3.11	31.94	94.00	6.00
Features: F_1, F_2, F_3						
5:5	93.78	92.66	7.34	6.22	93.22	6.78
4:6	91.22	94.08	5.92	8.78	92.94	7.06
3:7	88.17	95.63	4.37	11.83	93.39	6.61
2:8	85.42	96.33	3.67	14.58	94.15	5.85
1:9	60.27	98.37	1.63	39.73	94.56	5.44
Features: F_1, F_2, F_4						
5:5	88.28	93.00	7.00	11.72	90.64	9.36
4:6	87.13	92.80	7.20	12.87	90.53	9.47
3:7	86.53	93.80	6.20	13.47	91.62	8.38
2:8	83.73	94.41	5.59	16.27	92.28	7.72
1:9	76.18	96.23	3.77	23.82	94.22	5.78
Features: F_3, F_4						
5:5	92.82	94.88	5.12	7.18	93.85	6.15
4:6	91.53	94.41	5.59	8.47	93.26	6.74
3:7	90.56	95.49	4.51	9.44	94.01	5.99
2:8	87.41	95.88	4.12	12.59	94.19	5.81
1:9	79.16	97.46	2.54	20.84	95.63	4.37
Features: F_1, F_2, F_3, F_4						
5:5	96.87	93.50	6.50	3.13	95.19	4.81
4:6	95.88	93.39	6.61	4.12	94.38	5.62
3:7	94.21	94.94	5.06	5.79	94.72	5.28
2:8	89.85	95.46	4.54	10.15	94.34	5.66
1:9	92.12	96.36	3.64	17.88	94.94	5.06

in various simulated social network environments. Feature F_1 yielded a low TPR (13%) and a high TNR, comparing with other features. F_3 yielded a TPR of 51% and Features F_2 and F_4 demonstrated similar performance levels regarding the TPR and TNR. Overall, individual features exhibited inefficient detection, implying that multiple domain anomaly factors should be considered for improving the performance level.

Fig. 6 shows the experimental results when using multiple domain anomaly features in various simulated social network environments. The TPR might substantially degrade at a malicious to benign URL ratio of 1:9, because using few malicious test samples might affect the overall performance index. The results of feature set $\{F_1, F_2, F_3, F_4\}$ yielded the optimal level of performance among the combinations of features, indicating each feature is relevent for detecting malicious URLs.

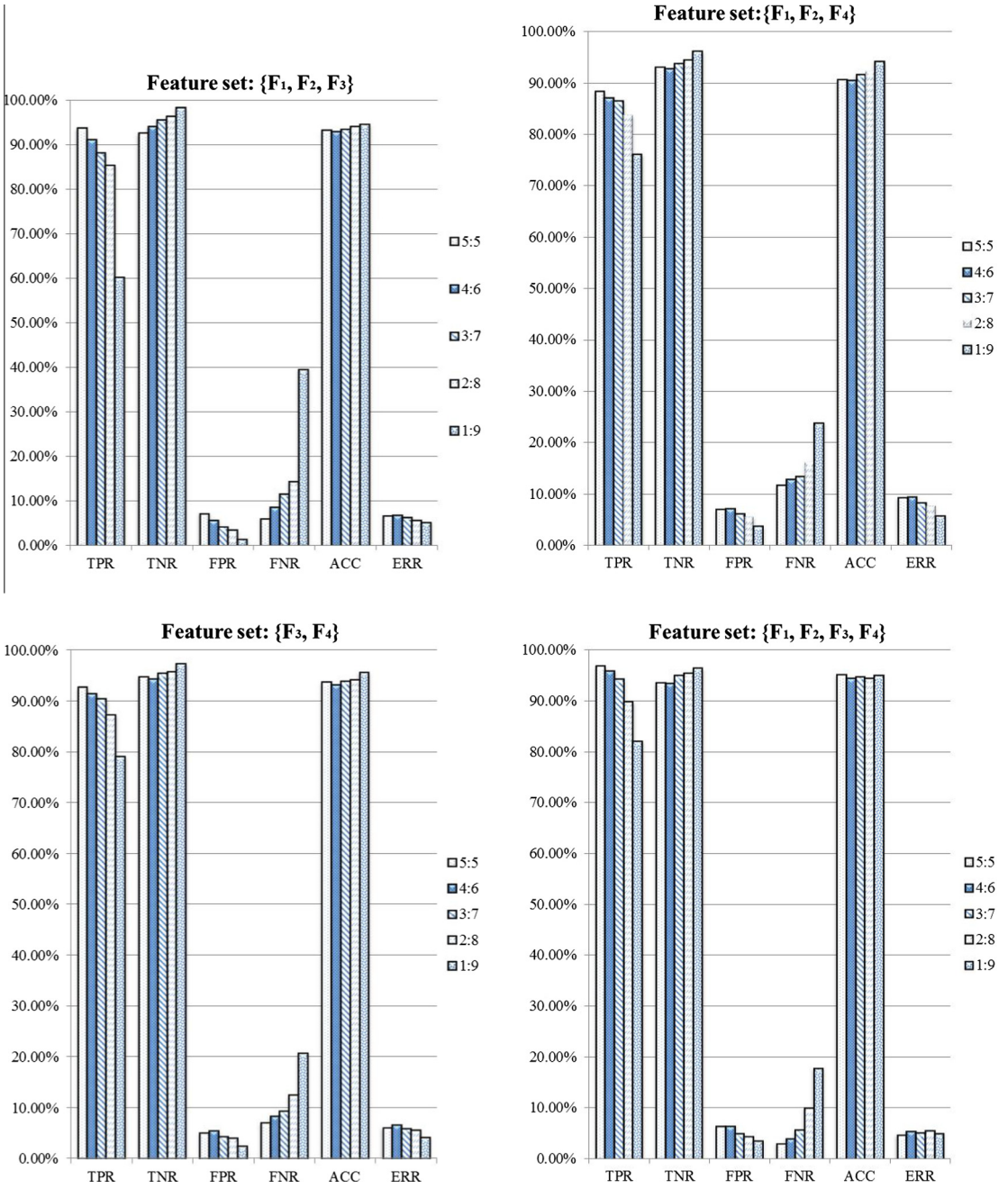

Fig. 6. The results of Experiment 1 with combinations of different domain anomaly features.

4.2.2. Experiment 2

Anomalous social behavior observed when posting may affect the level of detection performance. Two proposed social anomaly features F_6 and F_7 require setting a similarity threshold (Sim). The valid Sim range is 0–1 and a reasonable threshold should be larger than 0.5. A range of threshold values was evaluated to determine the sensitivity. The results are plotted in Fig. 7 and the numeric data is stated in Table 6. In contrast to Experiment 1, the proposed detection method yielded enhanced results when all domain and social anomaly features were applied. The results of Experiment 2 indicate that when a reasonable similarity threshold is employed, the proposed detection method performs efficiently and stably in various simulated social network environments. A low similarity threshold leads to a slightly low TPR, whereas a high threshold may slightly increase the FNR. The threshold variations are neglectable because most malicious web links exhibit high similarity values. A threshold value of Sim = 0.9, was used in subsequent experiments.

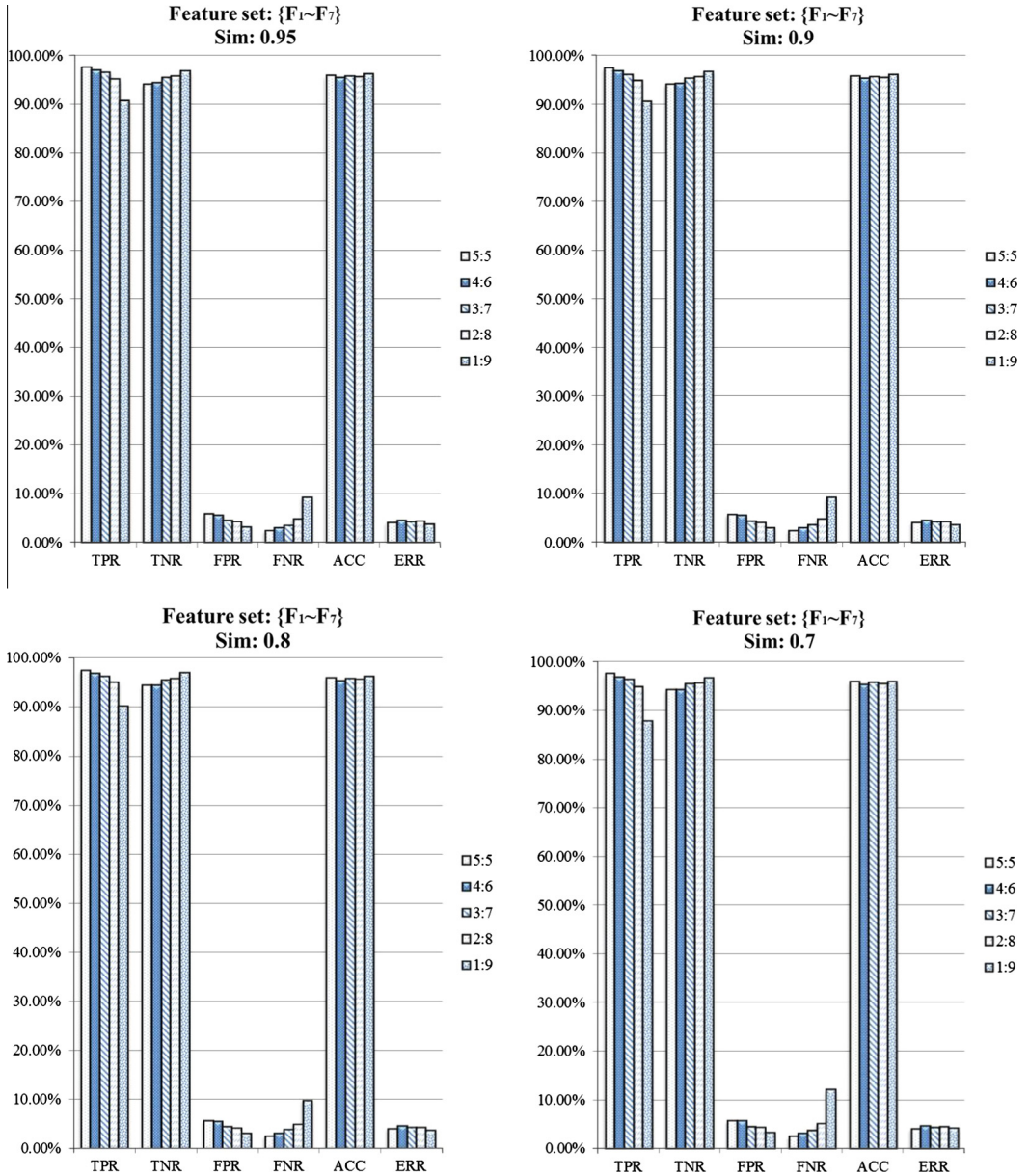

Fig. 7. The results of Experiment 2 with different similarity threshold values.

4.2.3. Experiment 3

Short URLs are used in various situations and could bias the proposed classification method. Therefore, the system performance was evaluated including and excluding short URLs. The world data set was evaluated, yielding 3976 malicious and 42,753 benign samples after excluding short URLs. The experiment results are shown in Fig. 8 and the numerical results are shown in Table 7. The results indicate that the proposed detection method can efficiently identify malicious URLs when short URLs are excluded or excluded; however, a superior performance level is obtained when excluding.

4.2.4. Experiment 4

Certain studies [1,13,32] have employed keyword-based or lexical features that might be sensitive to data from various geographic areas; thus, the collected keywords might be language dependent and might not be applicable to used in social networks in disparate language systems. To determine the sensitivity to various geographic social networks, the proposed detection method was evaluated using data collected from distinct geographic locations. Fig. 9 shows the evaluation results and Table 8 lists the numeric data. The proposed detection method exhibited adequate overall performance levels when

Table 6
Numeric results of Experiment 2.

M:B (%)	TPR (%)	TNR (%)	FPR (%)	FNR (%)	ACC (%)	ERR (%)
Features: all; Sim = 0.95						
5:5	97.57	94.07	5.93	2.43	95.82	4.18
4:6	96.99	94.29	5.71	3.01	95.37	4.63
3:7	96.44	95.44	4.56	3.56	95.74	4.26
2:8	95.07	95.70	4.30	4.93	95.57	4.43
1:9	90.73	96.74	3.26	9.27	96.14	3.86
Features: all; Sim = 0.90						
5:5	97.56	94.17	5.83	2.44	95.86	4.14
4:6	96.94	94.34	5.66	3.06	95.38	4.62
3:7	96.25	95.45	4.55	3.75	95.69	4.31
2:8	95.04	95.76	4.24	4.96	95.62	4.38
1:9	90.68	96.85	3.15	9.32	96.23	3.77
Features: all; Sim = 0.8						
5:5	97.47	94.34	5.66	2.53	95.91	4.09
4:6	96.83	94.40	5.60	3.17	95.38	4.62
3:7	96.15	95.51	4.49	3.85	95.70	4.30
2:8	95.00	95.76	4.24	5.00	95.61	4.39
1:9	90.17	96.93	3.07	9.83	96.25	3.75
Features: all; Sim = 0.7						
5:5	97.50	94.25	5.75	2.50	95.88	4.12
4:6	96.81	94.28	5.72	3.19	95.29	4.71
3:7	96.28	95.42	4.58	3.72	95.68	4.32
2:8	94.83	95.65	4.35	5.17	95.48	4.52
1:9	87.84	96.70	3.30	12.16	95.81	4.19

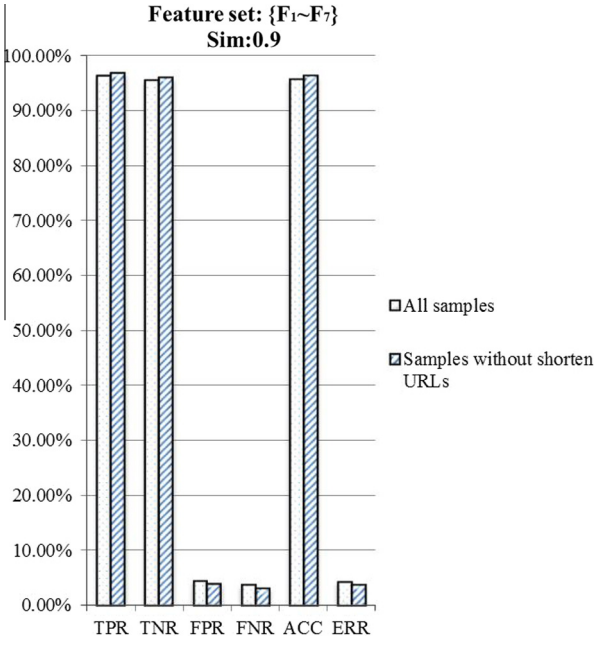

Fig. 8. The results of Experiment 3.

being used to assess various geographic social network data. The detection was slightly superior when assessing the world data set, and nonsignificantly degraded when evaluating the Asia and Europe data sets. The degradation was caused by unevenly sized test data sets; in other words, the data sets for Asia and Europe were much smaller compared with that of the world data set. Based on the findings of the four experiments, the proposed detection method yields superior detection rates when evaluating malicious and benign URLs.

Table 7
Numeric results of Experiment 3.

Features: all; Sim = 0.90						
Samples	TPR (%)	TNR (%)	FPR (%)	FNR (%)	ACC (%)	ERR (%)
All	96.25	95.45	4.55	3.75	95.69	4.31
No shorten URLs	96.82	95.97	4.03	3.18	96.23	3.77

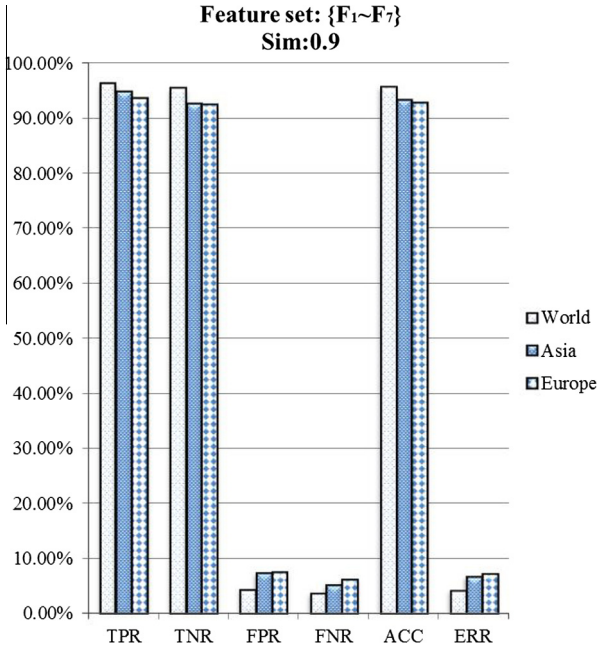

Fig. 9. The results of Experiment 4.

Table 8
Numeric results of Experiment 4.

Features: all; Sim = 0.90						
Samples	TPR (%)	TNR (%)	FPR (%)	FNR (%)	ACC (%)	ERR (%)
World	96.25	95.45	4.55	3.75	95.69	4.31
Asia	94.74	92.54	7.46	5.26	93.20	6.80
Europe	93.65	92.35	7.65	6.35	92.74	7.26

5. Conclusion

Blacklist mechanisms are not reliable for blocking malicious URLs in social network environments. The proposed solution does not rely on blacklist or whitelist mechanism, but rather uses URL information and the social behaviors of users. Two types of anomaly features were proposed: domain anomaly and social anomaly features. Domain anomaly features are used to identify possible malicious domains based on lexical and reputation factors, whereas social anomaly features represent anomalous user behaviors in social communications. The experimental results indicated that both anomaly types are essential for detecting suspicious URLs in social network environments. The proposed detection method involved Bayesian classification and the experimental results indicated that the method yielded efficient performance levels in various social network environments, yielding ACC of 95.7% and TPR of 96.3%.

Certain social network attacks might involve comprehensive techniques, such as embedding malicious code into links, encrypting malicious code, redirecting URLs, or other obfuscation techniques. Thus, detection mechanism should be able to identify these evasion techniques. Furthermore, people use SNSs to share diverse information, such as documents, movies, music, games, and multimedia data, and malicious code can be hidden in any type of data to hinder detection. Additional attributes, such as content-related, should be adopted when identifying diverse obfuscation and evasion techniques.

In this study, the data were collected from Facebook. Various ratios of malicious to benign sample URLs were selected to simulate various social network environments. However, the posts or feeds used on other social network services might differ from those used on Facebook. Data sets from other social networks should be examined to evaluate the robustness of the proposed detection method in various social network environments.

References

[1] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, Suku Nair, A comparison of machine learning techniques for phishing detection, in: Proceedings of the Anti-Phishing Working Group eCrime Researchers Summit, 2007.
[2] Jonell Baltazar, Joey Costoya, Ryan Flores, The Real Face of Koobface: The Largest Web 2.0 Botnet Explained, Technical report, Trend Micro Threat Research, 2009.
[3] André Bergholz, Gerhard Paaß, Frank Reichartz, Siehyun Strobel, Jeong-Ho Chang, Improved phishing detection using model-based features, in: Proceedings of the Conference on Email and Anti-Spam (CEAS), 2008.
[4] Facebook, Facebook Pages Statistics, Technical report. <http://statistics.allfacebook.com/pages> (accessed 2013).
[5] Facebook, Facebook Pages Statistics – Socialbakers, Technical report. <http://www.socialbakers.com/facebook-pages/> (accessed 2013).
[6] Facebook, Facebook Press Room, Technical report. <http://www.facebook.com/press/info.php?statistics> (accessed 2013).
[7] Facebook, Top Facebook Pages, Worldwide Social Media Stats, Technical report. <http://www.famecount.com/facebook-rank> (accessed 2013).
[8] Ian Fette, Norman Sadeh, Anthony Tomasic, Learning to detect phishing emails, in: WWW '07: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 649–656.
[9] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, Alfredo De Santis, Network anomaly detection with the restricted boltzmann machine, Neurocomputing (2013).
[10] D.J. Guan, Chia-Mei Chen, Jia-Bin Lin, Anomaly based malicious URL detection in instant messaging, in: Proceedings of the Joint Workshop on Information Security (JWIS), 2009.
[11] Xin Jin, Cindy Xide Lin, Jeibo Lui, Jiawei Han, A data mining-based spam detection system for social media networks, in: Proceedings of the VLDB Endowment, 2011.
[12] Gregg Keizer, Worm Spreads on Facebook, Hijacks Users' Clicks, Technical report, 2008. <http://www.computerworld.com/s/article/9122724/Worm_spreads_on_Facebook_hijacks_users_clicks>.
[13] Pranam Kolari, Tim Finin, Anupam Joshi, SVMs for the blogosphere: Blog identification and splog detection, in: Proceedings of the AAAI Spring Symposium on Computational Approaches to Analysing Weblogs, 2006.
[14] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker, Beyond blacklists: learning to detect malicious web sites from suspicious URLs, in: KDD '09 Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.
[15] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker, Identifying suspicious URLs: an application of large-scale online learning, in: Proc. of the International Conference on Machine Learning (ICML), 2009.
[16] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker, Learning to detect malicious URLs, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011).
[17] Mary Madden, Older Adults and Social Media, Technical report, Pew Internet & American Life Project, 2010. <http://pewinternet.org/Reports/2010/Older-Adults-and-Social-Media/Report.aspx>.
[18] D. Kevin McGrath, Minaxi Gupta, Behind phishing: an examination of phisher modi operandi, in: Proc. of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2008.
[19] Trend Micro, Trend Micro Corporate end User Survey: Global Rise in Workplace Social Networking. Technical report, 2010. <http://trendmicro.mediaroom.com/file.php/179/Trend+Micro+2010+Corporate+End+User+Study+-+PR2.zip>.
[20] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, Bobby Bhattacharjee, Measurement and analysis of online social networks, in: Proceeding IMC '07 Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, 2007.
[21] Cheryl D. Morse, Haining Wang, The structure of an instant messenger network and its vulnerability to malicious codes, in: Proc. of ACM SIGCOMM, 2005.
[22] Francesco Palmieri, Ugo Fiore, Network anomaly detection through nonlinear analysis, Comput. Secur. (2010).
[23] Irina Rish, An empirical study of the naive bayes classifier, in: Proceedings of IJCAI-01 Workshop on Empirical Methods in AI, 2001, pp. 41–46.
[24] Michael Robertson, Yin Pan, Bo Yuan, A social approach to security: using social networks to help detect malicious web content, in: 2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE), 2010.
[25] SPIEGEL Staff, Documents Reveal Top NSA Hacking Unit. Technical report, Spiegel Online, 2013. <http://www.spiegel.de/international/world/the-nsa-uses-powerful-toolbox-in-effort-to-spy-on-global-networks-a-940969.html>.
[26] Gianluca Stringhini, Christopher Kruegel, Giovanni Vigna, Detecting spammers on social networks, in: Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC 2010), 2010.
[27] Kurt Thomas, David M. Nicol, The koobface botnet and the rise of social malware, in: Malicious and Unwanted Software (MALWARE), 2010 5th International Conference, 2010.
[28] Dan Tynan, Facebook Botnets Have Gone Wild. Technical report, 2012. <http://www.itworld.com/it-managementstrategy/278005/faking-it-facebook-profile-bot-network>.
[29] Jaikumar Vijayan, Dhs Warns of Spear-Phishing Campaign Against Energy Companies, Technical report, ComputerWorld, 2013. <http://www.computerworld.com/s/article/9238190/DHS_warns_of_spear_phishing_campaign_against_energy_companies?taxonomyId=82>.
[30] Websense, Websense 2010 Threat Report, Technical report, Websense, 2010. <http://www.websense.com/content/threat-report-2010-social-networking.aspx>.
[31] Wei Xu, Fangfang Zhang, Sencun Zhu, Toward worm detection in online social networks, in: Proceeding ACSAC '10 Proceedings of the 26th Annual Computer Security Applications Conference, 2010.
[32] Yue Zhang, Jason Hong, Lorrie Cranor, CANTINA: a content-based approach to detecting phishing web sites, in: Proceedings of the International World Wide Web Conference (WWW), 2007.