A URL Address Aware Classification of Malicious Websites for Online Security during Web-surfing

Goutam Chakraborty, Tsai Tzung Lin
Department of Software & Information Science
Iwate Prefectural University, Japan
Email: goutam@iwate-pu.ac.jp, j2233223@gmail.com

Abstract—**On the Internet, users often visit unknown websites. However, malicious websites are a significant threat to the Internet users. Malicious websites implant malwares into users computers without their knowledge, through drive-by-downloads technology. A naive user could easily fall victim of such attack. With increased use of internet browsing, web security is an important issue and an important research topic. The motivation of this study is to classify malicious web-sites from benign ones from their URL features. If it could be done with high precision, especially with low false accept rate, automatic blocking of suspicious URL at the user site will be possible. We collect URL data for a large number of known benign as well as malicious websites. URL data has many characteristics. Some of them are relevant to classify the site as malicious and others are not. Many characteristic features are textual. We first converted all such features into suitable numeric data relevant to that feature, so that the numeric values truly represent the information of the original feature. We then select the relevant features for our classification task, by using two methods: (1) least absolute shrinkage and selection operator (LASSO) and (2) Multi-objective Pareto Genetic algorithm (MOGA). Finally, the data consisting of selected features is used to train a support vector machine (SVM) classifier. A ten-fold validation is used to estimate the performance. Performances of two feature selection methods as well as another recently published report were compared. By feature selection using Pareto GA, we could achieve more than 95% classification accuracy and F-score with least number of features.**

Index Terms—**URL address features, Malicious web-sites, Feature selection, LASSO, MOGA, SVM classifier,**

I. Introduction

Number of websites is growing exponentially because of a massive growth in web applications and services, such as social networking, blogs, and e-commerce. The internet has already become an important part of our daily life. Because of high-speed internet connection and wireless hot-spots available everywhere, the popularity of the Internet has naturally attracted miscreants. World Wide Web has become a platform to support a wide range of internet criminal activities such as spam-advertising, financial fraud, and malware implanting [3] [8] [9] [13] [15]. They set up various types of malicious websites to bait their victims. Though motivation and activities are different, the common target is to attract careless users to these fake websites which can be accessed via links through email, web search result, or links from websites redirection. All of these require the user to click on the so-called Uniform Resource Locator (URL). Thus, each time when the users decide to access unknown websites or click on an unfamiliar URL, they must perform sanity checks to pay attention to website address and use their intuition or experience to evaluate the associated risk that might be encountered. However, for the novice users, the chance of getting connected to malicious websites is high.

Many kinds of security technologies have been developed to escape these illegal events to occur. The most common and popular technology is to construct a blacklist, to protect the users from the phishing websites or drive-by-download attacks, by marking these malicious websites as dangerous [18] [19]. Blacklist has a basic access control mechanism to limit users to access websites whose information, such as IP address, URL, and domain, are in the blacklist. Blacklists are distributed to other users who use the technology to block malicious or phishing websites. Recently, blacklisting can be done by on-line application plug-in such as Google Blacklist and Bright Cloud or by Anti-virus software such as AVAST and McAfee. However, many malicious websites are not in the blacklist because they are created too recently or never evaluated. Blacklist cannot filter unknown malicious websites or new ones which are not contained in the blacklist.

In this work, we classify a website as malicious or benign based on its URL features. There are many features of which some are relevant for this classification task, and others are not. We need to focus on selecting significant features. Otherwise, irrelevant features which would act as noise, would reduce classification accuracy if included in the classification task. In the beginning, we consider all possible features. From URL, we adopted lexical features as well as host-based features, which characterize an URL. We use those features for classification.

This manuscript is organized as follows. Section 2 surveys related works. Section 3 is about data collection and feature extraction. The details of features and preprocessing of feature data, how to convert text features to numeric values, are explained. In section 4, feature selection methods, namely Lasso and Multi-Objective Genetic Algorithms (MOGA) are discussed. A few lines of explanation is added about the classifier, Support Vector Machine (SVM). Section 5 presents the experimental results. Here, we use ten-fold classifier to evaluate the accuracy and F1-score of classification result. Section 6 is the conclusion and discussion about future extension

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

of this research.

II. Related Works

Drive-by-download attack [3] [5] is, by which, malicious websites inject malware onto users computers when users visit these websites. Provos et.al. [15] identified four major types of the attack: advertising, third-party widget, web security application, and user contributed content. In drive-by-download attack, when a user browses the landing site, he will be directed to a drive-by-download server, usually called hop point. The hop point will identify the vulnerabilities of the user system and select the weakest one to launch an attack. The attack will command the browser to download malware from the malware distribution site. Finally, the malware is installed and executed automatically without user noticing it. Because the attack grows rapidly, any effective way to prevent the attack is to develop detection mechanism, before it is activated. The best way is to identify and refrain from connecting to malicious sites.

Blacklisting is a popular and widely used technology. Google blacklists approximately 9500 to 10000 websites per day [18]. However, though blacklisting prevents lots of malicious attacks, it is not effective to protect when the attacking websites are unknown. Crawler based searching and detecting malicious websites throught the whole internet is impossible. Sandboxing (testing on a different platform that will not affect the main system) is an effective way to detect a malicious website [21]. Yet it takes at least tens of seconds to verify a single site. Blacklisting can be combined with other technologies for better security. We propose to use machine learning to classify malicious websites in real-time before going to access an unknown website [11].

Ma et al. [8] used four datasets and validated the possibility of identifying malicious websites by using three machine learning models: Naive Bayes, Support Vector Machine with an RBF kernel and regularized logistic regression. Kazemian and Ahmed [6] compared several machine learning models including three supervised classifiers: K-Nearest Neighbor, SVM, and Naive Bayes; and three unsupervised techniques: Mini Batch K-Means, Affinity Propagation and K-Means. Supervised techniques could achieve a classification accuracy of 85-97%. Darling et. al. [12] developed a classification systems based on lexical analysis. They collected their datasets by configuring their crawler to collect from six sources and used 87 features for their decision tree based system. However, the main disadvantage was the fact that they used enormous number of features to achieve their results, making the process slow to train the decision tree. Finally, we will compare our results with that obtained by Darling.

In machine learning, reduction of features/factors is important because training would be computationally more efficient and improve the classification results by eliminating irrelevant factors which often times acts as noise. Feature selection is used when the data contain many features which are redundant or irrelevant. Thus they need to be removed for faster and better classification. In contrast to dimensionality reduction (like PCA), feature selection try to find a subset of the original variables. The goal of feature selection is to select the most significant features from the original feature set to construct a classifier that obtain the best performance.

Feature selection method is typically categorized into three classes: Filter Method, Wrapper Method, and Embedded Method. In Filter Method, features are evaluated using some statistical method and selected according to their scores. Filter methods recommend elimination of the least interesting features. The remaining features subset is used to classify or predict data. Filter methods do not consider what classifier will be used, and are mainly used in pre-processing stage because they do not consider the relationship among features. Wrapper methods consider the selection of a set of features as a search problem. Features are evaluated using a classifier model which can be a regression model, a K-nearest neighbor (KNN) classifier, a neural network, and so on. The search process may be methodical, stochastic, or heuristics to add and remove features. Embedded Method tries to combine the advantages of both Filter Method and Wrapper method. The most common type of embedded methods are regularization methods. In our experiment we use one of the embedded methods, LASSO, to find the significant features. In LASSO, regression model is used as classifier. Next, we used multi-objective optimization genetic algorithm (MOGA) to find the optimum feature set. The two objectives are minimizing the number of features and maximizing classification accuracy (or F-score).

The aim is to use feature selection to select the significant features based on lexical features and host-based features and evaluate the classification ability by using only the selected significant features. Results are compared for features selected by LASSO and MOGA, by evaluating classification accuracy using support vector machine classifier.

III. Data Collection and Feature Extraction

We collect a set of benign URL and malicious URL. Their URL features are listed. Features are to be preprocessed to numeric values to be suitable for classifier. Different feature selection methods were experimented to find the minimum set of features that could perform maximum accurate classification.

A. Data Collection

We collected data from the following sources: Clean mx [14] for the malicious websites which archives manually verified malicious URLs. We used Open Directory Project (DMOZ) [4] for the benign websites which contain user submitted URL and is the largest directory of the Web. Our input data have 46 thousand unique URLs which include 35 thousand benign URLs and 11 thousand malicious URLs.

B. Extracting Features and preprocessing of feature data

After data collection, we use relevant open source programs by python, such as python-whois, dnspython, ipwhois and so on, to collect URL features which are later used to categorize

URLs. They are either lexical or host-based. We use these features for classification, as is suggested to be important in the previous studies by McGrath et al. [3], Ma et al. [8], and Choi et al. [5].

Lexical feature

Lexical features are textual parts of URLs which allow the user to see the differences by reading the text. The URL has three main parts: the protocol, hostname, and path. For example, let us take the following URL:
http://www.iwate-pu.ac.jp/information/. The protocol is http://, the hostname is www.iwate-pu.ac.jp, and the path is /information/. Lexical features are the properties of the URL itself and do not include content of the web page [11]. Table. I shows lexical features and their description.

TABLE I
LEXICAL FEATURES

Features	Description
URL length	Length of Uniform Resource Locator
Domain length	Length of host name
Domain token count	Number of tokens in the hostname (delimited by '.')
Average domain token length	Average of domain token length
Longest domain token length	The longest domain token length
Dashes in domain name	Number of 'dash'es in domain
Numeric ratio	#Numerals in the domain name
Path Length	Length of the path name
Average path token length	Average of path token length
Longest path token length	The longest path token length
symbols in path	Number of symbols like '', '?', '+', '%', '=' etc.
Numeric ratio of Path	#Numerals in path namepath length

Host-based Features

Host-based features are used to find the hosting of the website and the reputation of the hosting center. The properties of hosts include IP address properties, geographic properties, domain name properties like Autonomous System Number (ASN), Country of ASN, Rough-index of a source, Number of name server, Rouge index of name server, Date of domain creation, Average path token length, Domain update, i.e., the last time the WHOIS information was either refreshed or changed.

C. *Preprocessing of the features*

We divide preprocessing of features into three categories: normalization of URL_length (real), Roguishness of a URL (real between 0 to 1), and feature from other data (of two categories, 0 or 1).

Normalization of data

We quantified and normalized data. Continuous variables were normalized to the range [0, 1]. We find the range that covers 97.5% of the data. Then, we linearly convert the range to "0 to 1" scale. Let us explain with URL length feature as an example. URL length distribution is with a sharp peak and a long asymptotic tail. The minimal URL length is 15 and the maximum URL length is 2821. Most of the data are located in the length from 15 to 163 (it included 97.5% data). The data whose URL length above 163 is only 2.5%. Therefore, we clipped the length at 163. Any URL length above that is mapped to the same value as URL of length 163, the longest.

We use the following formula to linearly convert this feature, to normalize its value, from 0 to 1. For all URL_length above 163 are mapped to 163, which is Max_URL.

$$Normalized\ Value = \frac{URL\_length - Min\_URL}{Max\_URL - Min\_URL}$$

The following features were normalized to a real value from 0 to 1: URL length, Domain token count, Average domain token length, Longest domain token length, #Dashes in domain name, Numeric ratio, Path Length, Average path token length, Longest path token length.

Rogue-index of a feature

The rogue-index of a feature is an estimate of the website to be benign or malicious. We calculate Rogue-index of a name server as is explained below. Name server record is relative to domain register. Many domain name registration agents offer very cheap service. Hacker may buy many domain name at low cost and perform malicious operations. Table. IV shows the name server which is used by more than 500 web-sites. We can see some name servers are more often used by malicious websites. We can calculate the roguishness index of a name server by the following formula.

$$Rogue - index = \frac{\mu/M}{\mu/M + \beta/B}$$

where, M is the total number of malicious sites ($\approx 11,000$ in our case), and μ are those with the feature under consideration (column 3 entry of Table IV). Thus, μ/M is the fraction of all malicious sites with that feature. Similarly, β is the number of benign sites with the feature under consideration. B is the total number of benign sites ($\approx 35,000$ in our case). By rogue-index of a name server, we estimate the probability whether it is a Rogue or not. If the rogue-index is close to 1, the name server is possibly used by malicious websites, and if the roguishness index is close to 0, the name server is possibly operated by a benign domain registration agent. An example for "Name Server" feature is shown in the table IV. The rouge-index is calculated based on the cardinality of our data-set, where $M \cong 11,000$ and $B \cong 35,000$.

TABLE II
EXAMPLES OF ROGUE INDEX FOR NAME SERVER FEATURE

Name Server	#benign	#malicious	#rouge index
DOMAINCONTROL.COM	1381	868	0.6699
LYCOS.COM	1273	1	0.0025
COM.BR	14	641	0.9933
CLOUDFLARE.COM	494	148	0.4917
HOSTGATOR.COM	303	338	0.7827
DOITBROTHER.COM	0	615	1.0000
WORLDNIC.COM	586	24	0.1168
SUPERDNSSITE.COM	0	601	1.0000
CO.UK	387	197	0.6217
DREAMHOST.COM	542	22	0.1159

Other Features

TLD features are ccTLD (Country Code top-level domain) and gTLD (Generic top-level domain). The value is equal to

“1” for ccTLD and 0 for gTLD.

Domain create date/last update date

This is a date data, from which we create the value, how old the site is, in unit of months. 1970 is considered to be base year, as it is the oldest site in our database. An old site is more likely to be benign compared to a very new site. We need to convert the date (included year and month) into a value and then convert the value to normalized value. In our data, 1970/01 is the first month, so the value is normalized to 1. We use the following formula to convert date data to a numerical value:

$$date\_feature\_value = 12 \times (year - 1970) + month$$

IV. Feature Selection and Classification

In this step, we use LASSO and genetic algorithm for feature selection, to find significant features. Lasso is least absolute shrinkage and selection operator. It was introduced by Robert Tibshirani in 1996 based on Leo Breimans Nonnegative Garrote [10, 16]. GA is a well known general and robust search algorithm based on the idea of *survival of the fittest.*

A. LASSO

LASSO is widely used to select significant feature subset. Among existing feature selection algorithms, LASSO has been accepted as efficient and the most practical one because of its low computational cost, robustness and good precision [7, 20]. It can automatically select the number of variables by shrinking the coefficient values of variables and setting some equal to zero. Besides, LASSO coefficient values determine the importance of different factors. The more its absolute value is, the more important the factor is. The LASSO method does minimize the following argument

$$(\alpha, \beta) = \begin{array}{r} arg\ min \sum_{t=1}^{T} (y_t - \beta_i x_{i,t} - \ldots - \beta_k x_{k,t})^2, \\ subject\ to\ \sum_{j=1}^{k} |\beta_j| \leq \lambda \end{array}$$

where, T is the number of data, y_t is dependent variable, k is the number of independent variables, $x_{i,t}$ are independent variables, λ is the tuning parameter, β_i the regression coefficient. The tuning parameter $\lambda \geq 0$ controls the amount of shrinkage applied to the estimates.

B. Feature selection by genetic algorithm (GA)

We used Pareto Genetic algorithm to find optimal feature subset. For chromosome fitness we use Support vector machine to find classification accuracy, using the selected subset of features. Pareto Genetic algorithm (Pareto GA) solves multi-objective problems using Genetic algorithm (MOGA). In this experiment, we have two objectives: less number of features and higher classification accuracy.

Crossover

In genetic algorithm, crossover is used to combine chromosomes to get new solution from one generation to the next generation. It is similar to reproduction in biological crossover. Crossover takes two parent chromosome and swap part of them to generate two children from the parents chromosomes. We use 2-point crossover and the crossover rate was 100%.

Mutation

Mutation alters one or more gene values in a chromosome. The advantage is to get a new chromosome and avoid local minimum by preventing the population of chromosomes from becoming too similar. In our study, the mutation probability is decreased with the number of generation, starting with a high value of 0.05 to get varieties of chromosomes (higher exploration), and mutation rate is decreased in steps to 0.01.

Fitness Evaluation and Tournament Selection In this step, we evaluate a chromosome by considering two indicators: the number of features and classification accuracy. In each generation, we normalize the result of both and add the two values. We use tournament selection to select chromosome for the next generation.

C. Training and classification by SVM

We used Support vector machine (SVM) classifier with rbf kernel to evaluate the classification accuracy. The classification result is evaluated using five performance evaluation criteria including accuracy, precision, recall, F1-score, and training time. In the experiment, 10-fold cross validation is applied to the data set. In 10-fold cross validation [17], the data set is divided into 10 random partitions. One-tenth is used for testing the classifier and nine-tenth is used for training the classifier. The training and testing data are changed and classification performance is calculated for 10 different sets of testing data. The average value is given.

We have an unbalanced data set, with benign data around 35,000+ and malicious data about one-third of it. In experiment 1, we use the whole dataset and evaluate the result by Precision, Recall, F1-Measure, Training Time, and Accuracy.

In experiment 2, we divide the benign websites into three subsets (each subset consists of a little more than 11000 records) to let the volume of both (malicious and benign) data to be the same (or similar). For each subset, we re-calculate the rogue-index value, perform feature selection and classification separately and evaluate the result.

V. Experimental Results

A. Unbalanced Data and Feature selection by LASSO

In Experiment 1, unbalanced data is used, where begin samples are 3 times more compared to malicious ones. Table. V shows the result of significant features selected by LASSO. We use descending order of coefficient absolute values (without considering the sign, positive or negative) to sort the features. In the result, first 16 are the significant features with non-zero coefficients, and last 5 are features with coefficient values equal to zero.

Significant features selected by LASSO are used for classification between benign and malicious sites. We use SVM to evaluate the performance. We select different numbers of significant features from 5 to 16 by descending order of the feature coefficient values and evaluate the accuracy of the classifier, and noted the required training time.

Table. VI shows the classification accuracy and computation time. In every row, we packed 3 results with 3 consecutive

TABLE III
RESULTS OF LASSO FEATURE SELECTION

Feature	Lasso coefficint	Feature	Lasso coefficint
Rogue of NS	-1.087	Longest path token length	-0.047
Domain token count	0.918	Av. domain token length	-0.044
ASN	-0.621	Domain length	-0.040
Country of ASN	-0.173	Longest domain token len	-0.020
TLD type	-0.129	Rogue of source	0.001
Path length	-0.077	Dash in domain	0
Av. path token length	-0.076	Numeric ratio of path	0
URL length	-0.064	Country	0
Symbols in path	-0.050	Domain create date	0
Numeric ratio of dom	-0.049	Last update date	0
Name server amount	-0.048		

number of features. We can see that even with as low as 5 features we can get a good classification accuracy at a low computation cost.

TABLE IV
SELECTED FEATURES AND CORRESPONDING CLASSIFICATION ACCURACY

Number of Features	Classification accuracy	Time(sec) required for Training & Testing
21 (all features)	95.9	66 seconds
16, 15, 14	95.8, 95.8, 95.8	60, 52, 49
13, 12, 11	95.7, 95.6, 95.6	49, 47, 44
10, 9, 8	95.7, 95.5, 95.4	42, 42, 43
7, 6, 5	95.4, 95.1, 95.1	41, 43, 43

Table. VII shows the classification result for three different number of features: (i) all features, (ii) significant features selected by LASSO (with non-zero coefficients), and (iii) 10 features with highest LASSO coefficients. We can see all the classification accuracy are above 95.7% and F1-Measure are above 90%. Even with 10 features, the classification accuracy is not degraded significantly.

In our experiment, the benign websites accuracy is above 97% and the malicious websites accuracy is much less, only above 90%. The reason is because of the unbalance in data sizes.

TABLE V
SELECTED FEATURES AND CORRESPONDING CLASSIFICATION ACCURACY. PRECISION, RECALL, F1-SCORE, ACCURACY ARE IN %, TIME IN SECONDS

Item	Precision	Recall	F1	Accuracy	Time
21 features (all features)					
Benign	96.7	98.0	97.3	95.9	66
Malicious	93.2	89.1	91.1		
16 features (features with non-zero Lasso)					
Benign	96.7	97.9	97.3	95.82	60
Malicious	92.9	89.0	91.0		
Top 10 features by Lasso					
Benign	96.4	97.8	97.1	95.7	42
Malicious	92.7	88.2	90.4		

B. Results with balanced data

In experiment 2, we balance the data from two classes to improve the classification accuracy. We divided the benign dataset into three subsets by random and unique selection, by which we mean data in three subsets are unique. The number of benign data in three subsets are similar to the malicious dataset.

For three different subsets of the benign dataset, and the common malicious web-site data set, Lasso feature selection is performed separately for 3 times. For three different sub-sets the top 10 selected features have mostly common members. The features with 0 coefficient were different for different subsets, 5 for subset I, and 7 for subset II and 8 for subset III. The 8 features with zero coefficient for subset III includes all 7 rejected features of subset II, and they both include the 5 features with zero coefficients from subset I. For lack of space, we are not giving the detail results. The order in which lasso coefficient appear, for the 3 different subsets of data, are very similar. When we use balanced dataset from both classes, precision and recall, for malicious web-sites improved. For benign web-websites it it is decreased a little to match the value obtained for benign data. As we divide the benign data in 3 subsets, we get 3 sets of results. The values of precision and recall of all sets are quite similar. Result with subset 1 is shown in Table.VIII.

Results with subset 1 benign data

Table. VIII shows the results of Experiment 2 with benign data subset 1. Accuracy with all 21 features is 95.56%, with 16 features (those with non-zero lasso coefficients) is 95.59%, and with top 10 features is 95.35%. All classification criteria (precision, recall, F1-score) were above 95%. Results for both benign and malicious websites are similar. Compared to unbalanced data, though the classification results are a little worse for benign data, it improved significantly for malicious data.

TABLE VI
CLASSIFICATION ACCURACY FOR BALANCED DATA. PRECISION, RECALL, F1-SCORE, ACCURACY ARE IN %, TIME IN SECONDS

Item	Precision	Recall	F1	Accuracy	Time(sec)
21 features (all features)					
Benign	95.7	95.4	95.6	95.56	43
Malicious	95.5	95.7	95.6		
16 features (features with non-zero Lasso)					
Benign	95.7	95.4	95.6	95.59	27
Malicious	95.5	95.7	95.6		
Top 10 features by Lasso					
Benign	95.5	95.2	95.3	95.36	15
Malicious	95.2	95.5	95.4		

C. Comparison of results: Feature selection by Lasso and MOGA

We compare the results between LASSO and pareto GA. Table. IX summarizes the results of comparison. In feature selection, LASSO needs less time to decide the significant features. Pareto GA took around 10 hours to find the best features. This is because for evaluation of every chromosome, we need to run SVM to know the classification accuracy for that set of features. Pareto GA can achieve better result compared to LASSO because LASSO uses linear regression model, whereas in Pareto GA we used SVM for classification. The top 5 selected features are the same: Rogue-index of Name

Server (NS), ASN, Domain token count, Path Length, and URL length.

TABLE VII
COMPARISON OF RESULTS: FEATURES SELECTED BY LASSO AND MOGA

Item	LASSO+SVM	Pareto GA+SVM
Training time	5 minutes	10 hours
Accuracy Top 5 features Top 10 features	 93.82% 95.28%	 95.11% 95.50%
Selected Features	*Rogue of NS* *ASN* *Domain token count* *Path length* *URL length* Numeric ratio of path Av. domain token length Rouge of source	*Rogue of NS* *ASN* *Domain token count* *Path length* *URL length* TLD Name server number Longest path token length Domain create date

D. Comparison with Darling's [12] work

Table. X shows the comparison of the previous related works and our results. Both of Darlings [12] and our datasets have same condition that all domains are unique. In our experiments, we use subset1 to compare with. For Lasso and Pareto GA, we used the number of features that achieved the best result, although even with less number of features the results degraded only slightly.

We can see our results are better than Darlings. There are two reasons: First, we use feature selection to eliminate the noisy and irrelevant features. In addition, we use SVM method which is considered more accurate than J48 used in Darlings work. Second, Darling used lexical features but we use lexical features and host-based features. Besides, the feature rogue-index of name sever, is a significant feature as we see. That was absent in Darlings work.

TABLE VIII
COMPARISON OF RESULTS: PROPOSED METHODS AND DARLING'S WORK [12]

	Proposed	Proposed	Darling's
Feature Selection	by Lasso	by Pareto GA	No
Classification by	SVM	SVM	J48
Total features	21	21	87
#Selected features	16	13	87
Data size	21980	21980	12000
Best Accuracy(%)	95.59	96.05	94.76
F1-score(%)	95.6	95.4	94.7

VI. CONCLUSION AND FUTURE WORK

Conclusions: In Experiment 1, classification performances were all above 95% for benign websites, but for malicious websites precision and recall were much less. In Experiment 2, by balancing the benign and malicious website data set sizes, the classification performances were all above 95%. Though results were a little degraded for benign data, for malicious website data the recall improved by more than 6% points.

In Experiment 2, we find less number of features and get a better classification accuracy using pareto GA. Through comparison, we have shown that there are common significant features selected by both methods.

Future work: In future, we will design a system which would be plugged-in the user system to detect the websites and distinguish the malicious and benign websites in real-time on real internetwork. The system may train and distinguish automatically so that it can detect unknown websites on the fly.

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks". Machine Learning No. 20, pp.273-297 (1995).

[2] C. M. Chen, J. J. Huang, Y. H. Ou, "Efficient suspicious URL filtering based on reputation". Journal of Information Security and Applications Vol. 20, pp.26-36 (2015)

[3] D. K. McGrath and M. Gupta, "Behind Phishing: An Examination of Phisher Modi Operandi". Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET), SF, CA, (2008).

[4] DMOZ - The Directory of the Web. https://www.dmoz.org/ (online)

[5] H. Choi, B. B. Zhu, and H. Lee, "Detecting Malicious Web Links and Identifying Their Attack Types", WebApps'11, Proceedings of the 2nd USENIX conference on Web application development, pp.11-11, Berkeley, USA (2011)

[6] H. B. Kazemian and S. Ahmed, "Comparisons of machine learning techniques for detecting malicious webpages", Expert Systems with Applications, pp. 1166-1177 (2015).

[7] H. Xu, C. Caramanis, and S. Mannor, "Robust Regression and Lasso", IEEE Transactions on Information Theory, pp. 3561 - 3574 (2010).

[8] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs", Proceedings of the 15th ACM SIGKDD international conference on Knowledge discov ery and data mining, New York, USA, pp. 1245-1254 (2009).

[9] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and Evaluation of a Real-Time URL Spam Filtering Service", 2011 IEEE Symposium on Security and Privacy, USA, pp. 447-462 (2011).

[10] L. Breiman, "Better Subset Regression Using the Nonnegative Garrote", Technometrics No.37, Pp.373-384 (1995).

[11] M. Aldwairi and R. Alsalman, "MALURLS: A Lightweight Malicious Website Classification Based on URL Features", Journal of Emerging Technologies in Web Intelligence (JETWI) Vol. 4, pp.128-133 (2012)

[12] M. Darling, G.Heileman, G. Gressel, A. Ashok, and P. Poornachandran, "A Lexical Approach for Classifying Malicious URLs", High Performance Computing & Simulation (HPCS), USA, pp.195-202 (2015).

[13] M. Egele, E. Kirda, and C. Kruegel, "Mitigating driveby download attacks: Challenges and open problems", In Proceedings of Open Research Problems in Network Security Workshop (iNetSec 2009), Zurich, Switzerland (2009).

[14] Malware - Clean MX. http://support.clean-mx.de/clean-mx/viruses.php

[15] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadug, "The ghost in the browser analysis of web-based malware", In: Proceedings of the first workshop on hot topics in understanding botnets, Cambridge (2007).

[16] R. Tibshirani, Regression Shrinkage and Selection via the lasso, Journal of the Royal Statistical Society Vol.58, pp.267-288 (1996).

[17] S. Salzberg, "On comparing classifiers: pitfalls to avoid and a recommended approach", Data Mining and Knowledge Discovery, Boston, USA, pp. 317-328 (1997).

[18] Security: https://sucuri.net/website-security/google-blacklisted-my-website (online)

[19] Y. Fukushima, Y. Hori, and K. Sakurai,"Proactive Blacklisting for Malicious Web Sites by Reputation Evaluation Based on Domain and IP Address Registration", 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, Changsha, China, pp.352-361 (2011).

[20] Zhou, Q., Song, et.al., Efficient Lasso training from a geometrical perspective, Neurocomputing, Vol. 168, pp. 234-239 (2015).

[21] Ourston D, Matzner S, Stump W, Hopkins B., "Application of hidden Markov models to detecting multi-stage network attacks," In System Sciences, Proceedings of the 36th. Annual Hawaii Intl. Conf., 2003.