A Lightweight Online Advertising Classification System using Lexical-based Features

Xichen Zhang, Arash Habibi Lashkari and Ali A. Ghorbani
Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB), Canada

Keywords: Advertisement Detection, Machine Learning, Characterization, Lexical Features.

Abstract: Due to the significant development of online advertising, malicious advertisements have become one of the major issues to distribute scamming information, click fraud and malware. Most of the current approaches are involved with using filtering lists for online advertisements blocking, which are not scalable and need manual maintenance. This paper presents a lightweight online advertising classification system using lexical-based features as an alternative solution. In order to imitate real-world cases, three different scenarios are generated depending on three different URL sources. Then a set of URL lexical-based features are selected from previous researches in the purpose of training and testing the proposed model. Results show that by using lexical-based features, advertising detection accuracy is about 97% in certain scenarios.

1 INTRODUCTION

The exploding development of World Wide Web in the mid-1990s (Krammer, 2008) led to the birth of a new and profitable business model – online advertising. Then in the next following decades, due to the expansional growth of mobile devices such as laptops and smart phones, web-based contents and services had become more easily accessible, and online advertising industry started flourishing. Involving with the use of the Internet, online advertising is a marketing strategy that can deliver commercial messages and business information to the customers efficiently. In order to draw the attention of potential customers, enhance the brand recognition/awareness, and generate a favorable reputation for the products, web advertisers tend to design distinctive online advertisements constantly and display them in a wide variety of types. Unfortunately, as the rapid growth of online advertising, a massive amount of unnecessary and intrusive contents are downloaded during web surfing (Szczepański et al., 2013). Users cannot get access to the useful messages on the Internet effectively. And a majority of the customers consider online advertisements as annoying, obtrusive, distracting, and all over the places (Adobe, 2013) (Jover et al., 2015) (Andriatsimandefitra and Tong, 2014).

Furthermore, online advertising is now the predominant model for marketing and promotion, and with it online advertising fraud such as propagating malware, scamming, click fraud, etc. (Li et al., 2012), has become an increasing concern among researchers. Currently, there are potential interests in building systems to prevent customers from visiting these websites (Ma et al., 2009).

The main contribution of this paper is twofold. First, we present a URL-based approach for online advertisement classification. And based on previous studies, we propose an approach to select features with high distinguishing power for identifying online advertisements. Second, we select a new set of features for detecting online advertisements. With the application of machine learning techniques, our proposed features can provide efficient information for characterizing online advertisements, and can be automatically adapted to large datasets.

The rest of this paper is organized as follows: Section 2 gives the related works; Section 3 describes how to select the most indicating features; Section 4 presents how to collect the dataset; Section 5 gives the implementation of our experiment; Section 5.2 shows the analysis of the results; and Section 6 recaps the conclusion of this paper.

2 RELATED WORKS

Related works from 2004 to 2015 have been reviewed in order to identify a collection of commonly used features that could yield high detection accuracy. Table 1 shows 36 varied features extracted from the re-

Zhang, X., Lashkari, A. and Ghorbani, A.
A Lightweight Online Advertising Classification System using Lexical-based Features.
DOI: 10.5220/0006459804860494
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 486-494
ISBN: 978-989-758-259-2

Table 1: List of detection features fetched from previous works.

#	Category	Feature Name	Description	Objective	Reference
F1	Textual-based Feature	Textual length of URL	The total length of the URL	Detect ads or malicious ads	(Szczepański et al., 2013) (Kan and Thi, 2005) (Ma et al., 2009) (Ma et al., 2011)
F2	URL-based Feature	URL component presence	If query component or user information component present in the given URL	Detect ads	(Szczepański et al., 2013) (Kan and Thi, 2005)
F3	Textual-based Feature	Token occurrences	The occurrences of each word in URL	Detect ads or web content	(Szczepański et al., 2013) (Baykan et al., 2011) (Baykan et al., 2009) (Devi et al., 2007)
F4	URL-based Feature	Token occurrences by URL component	The count of token occurrences for each component	Detect ads	(Szczepański et al., 2013) (Kan and Thi, 2005)
F5	Textual-based Feature	Sequential n-gram	URL first split into tokens, then derive sequence of n tokens from them	Detect ads	(Baykan et al., 2011) (Szczepański et al., 2013) (Kan and Thi, 2005) (Baykan et al., 2009) (Zhang et al., 2006)
F6	Textual-based Feature	Full token n-gram	The count of occurrences of tokens with regard to succession relation	Detect ads	(Szczepański et al., 2013)
F7	URL-based Feature	Token count (total and per URL component)	Ads are likely to have many parameters in query component	Detect ads	(Szczepański et al., 2013)
F8	Textual-based Feature	Numeric tokens count (total and per URL component)	The count of occurrences of numeric values in URL	Detect ads	(Szczepański et al., 2013)
F9	Textual-based Feature	Ad-related keywords	Such as 'ad', 'advert', 'popup','banner', 'sponsor', 'iframe', 'googlead', 'adsys' and 'adser'	Detect ads or phishing URL	(Bhagavatula et al., 2014) (Krammer, 2008) (Pradeepthi and Kannan, 2014)
F10	Textual-based Feature	Suspicious symbol presence	If the URL contain semicolons to separate parameters? If the URL contains valid query? If any suspicious symbol just like @ present?	Detect ads or phishing URL	(Bhagavatula et al., 2014) (Pradeepthi and Kannan, 2014)
F11	URL-based Feature	Original page related	If the base domain name is present in the query of the URL? If the requested URL is on the same domain?	Detect ads	(Bhagavatula et al., 2014)
F12	URL-based Feature	Size of Ads in URL	Indicating the size of the ads it going to display	Detect ads	(Bhagavatula et al., 2014)
F13	URL-based Feature	Dimensions of the URL	Indicating the dimensions of the screen or browser	Detect ads	(Bhagavatula et al., 2014) (Shih and Karger, 2004) (Krammer, 2008)
F14	URL-based Feature	Iframe container	Indicating if the URL is requested from within an iframe either in the context of the page or in the context of nested iframes	Detect ads	(Bhagavatula et al., 2014)
F15	URL-based Feature	Proportion of external requested resources	The proportions of external iframe, script and resource requests	Detect ads	(Bhagavatula et al., 2014)
F16	Textual-based Feature	Textual length of hostname	The length of hostname in given URL	Detect malicious ads	(Ma et al., 2009) (Ma et al., 2011)
F17	Textual-based Feature	Dots occurrences	The number of dots in the URL	Detect malicious ads	(Ma et al., 2009) (Ma et al., 2011) (Pradeepthi and Kannan, 2014)
F18	Host-based Feature	IP address	Is the IP address in the blacklist?	Detect malicious ads	(Ma et al., 2009) (Ma et al., 2011)
F19	Host-based Feature	WHOIS properties	What is the date of registration, update and expiration? Who is the registrar? Who is the registrant? Is the WHOIS entry blocked?	Detect malicious ads	(Ma et al., 2009) (Ma et al., 2011) (Pradeepthi and Kannan, 2014) (Li et al., 2012)
F20	Host-based Feature	Domain name properties	What is the TTL for DNS records associated with the hostname?	Detect malicious ads	(Ma et al., 2009) (Ma et al., 2011)
F21	Host-based Feature	Geographic properties	In which continent/country/city does the IP address belong?	Detect malicious ads	(Ma et al., 2009) (Ma et al., 2011)
F22	JavaScript Feature	JavaScript source code	Including code obfuscation, dynamic code and URL generation, code structure, function call distribution, event handling, script origin, presence of keywords	Detect ads	(Orr et al., 2012) (Yu, 2015)
F23	URL-based Feature	URL tree	The left-most item (http:) becomes the root node of the tree, successive tokens in the URL become the children of the previous token	Detect ads or web content	(Shih and Karger, 2004)
F24	Host-based Feature	Blacklist membership	Is the IP address in a blacklist?	Detect malicious URL	(Ma et al., 2011)
F25	Host-based Feature	Connection speed	What is the speed of the uplink connection?	Detect malicious URL	(Ma et al., 2011)
F26	Textual-based Feature	Presence of IP address	IP is not included in a normal URL	Detect phishing URL	(Pradeepthi and Kannan, 2014)
F27	Textual-based Feature	Unknown noun presence	Domain names are not created by using some random letters	Detect phishing URL	(Pradeepthi and Kannan, 2014)
F28	URL-based Feature	Misplaced top level domain	If the domain name is present in the path section?	Detect phishing URL	(Pradeepthi and Kannan, 2014)
F29	Network-based Feature	URL redirection	If the URL has been used to redirect to many other pages?	Detect phishing URL or ads	(Pradeepthi and Kannan, 2014) (Yu, 2015) (Krammer, 2008) (Li et al., 2012)
F30	Network-based Feature	Traffic received	The amount of web traffic that each website gets	Detect phishing URL	(Pradeepthi and Kannan, 2014)
F31	Network-based Feature	HTML table tree	The visual/physical placement of links on a page	Detect ads or webpage content	(Shih and Karger, 2004)
F32	Textual-based Feature	Precedence Bigram	The left-to-right precedence of two tokens in the URL	Detect ads	(Kan and Thi, 2005)
F33	Network-based Feature	URL redirect path	The redirection chain of a set of URLs	Detect malicious ads	(Li et al., 2012)
F34	Network-based Feature	Domain redirect path	The redirection chain of a set of domains	Detect malicious ads	(Li et al., 2012)
F35	Host-based Feature	Domain frequency	The number of publishers that associated with the domain each day	Detect malicious ads	(Li et al., 2012)
F36	Host-based Feature	Domain-pair frequency	The frequency of two neighboring URLs/domains	Detect malicious ads	(Li et al., 2012)
F37	Textual-based Feature	Dash count in hostname	The number of dash present in hostname	Detect suspicious URLs	(Chen et al., 2014)

lated works along with short descriptions.

Lawrence Kai Shih and David R. Karger (Shih and Karger, 2004) in 2004 proposed a URL-based approach for webpage identification for ad-blocking and content recommendation. They used four training systems for data labeling. The four systems are: WebWasher, Redirect, Learn-WW and Learn-RD. And the two type of features they used are URL-based feature: $\{F_{23}\}$, and Network-based feature: $\{F_{31}\}$.

Min-Yen Kan and Hoang Oanh Nguyen Thi (Kan and Thi, 2005) in 2005 demonstrated the use of URL alone in performing webpage classification, such as identification of online advertising. They applied machine learning classifiers like Support Vector Machines (SVM) and Maximum Entropy (ME) to evaluate the performance of the following features: $\{F_1, F_2, F_4, F_5, F_{32}\}$

Jianping Zhang *et al.* (Zhang et al., 2006) in 2006, M. Indra Devi *et al.* (Devi et al., 2007) in 2007 and Eda Baykan *et al.* (Baykan et al., 2009) in 2009 stud-

ied the problem of URL-based webpage classification with machine learning techniques. With the application of SVM, Naive Bayes (NB) and ME, both of the authors broke the URL into a sequence of tokens, and treated the URL tokens as the feature $\{F_3, F_5\}$.
Hern *et al.* in 2014 (Hernández et al., 2014) and 2016 (Hernández et al., 2016) proposed a URL-based, unsupervised tool called CALA, for webpage classification. In their article, CALA took the URL of a webpage as input, and then output a set of patterns that represent the URL in different semantic classes.
Viktor Krammer *et al.* (Krammer, 2008) in 2008 introduced a web browser-based content filter — Quero. And by applying different rules and features, they presented how to use Quero against web advertising. The features related to URL included: $\{F_3, F_{13}, F_{29}\}$.
Justin Ma *et al.* (Ma et al., 2009) in 2009 detected malicious websites by automated URL classification. In their approach they measured different classification models such as NB, SVM and Logistic Regression (LR) with mainly two types of features, lexical-based features: $\{F_1, F_{16}, F_{17}\}$, and host-based features: $\{F_{18}, F_{19}, F_{20}, F_{21}\}$. (Ma et al., 2011) put forth a similar study in 2011, and evaluated the same features in (Ma et al., 2009). In addition, a real-time system was developed with online classifiers such as Perceptron, LR with Stochastic Gradient Descent, Passive-Aggressive Algorithm (PA) and Confidence-Weighted (CW) Algorithm.
Ludmila Marian *et al.* (Baykan et al., 2011) in 2011 classified webpage contents by URL analysis. The four main features they used can be categorized into two types: lexical-based feature and URL-based feature. The authors evaluated their approach by applying machine learning techniques such as NB, SVM, ME, and Boosting with features: $\{F_3, F_5, F_{23}, F_{24}\}$.
Caitlin R.Orr *et al.* (Orr et al., 2012) in 2012 implemented a static analysis-based approach to identify ad-related JavaScript. They assessed the performance of their approach by applying SVM with the features that extracted from given scripts: $\{F_{22}\}$.
In order to detect malicious web advertising, Zhou Li *et al.* (Li et al., 2012) in 2012 developed a topology-based system relied on the analysis of URL redirection chains. With network-based features $\{F_{29}, F_{33}, F_{34}\}$ and host-based features $\{F_{19}, F_{35}, F_{36}\}$, they adopted a statistical learning framework based on decision trees to automatically generate a set of detection rules for identifying malicious advertisements.
Piotr L. Szczepański *et al.* (Szczepański et al., 2013) in 2013 presented a URL-based automated framework as a solution to the problem of online advertisements detection. Their experiments were performed on the following popular classifier, K-Nearest Neighbour (KNN), NB, Bayesian Network, SVM, Decesion Tree, Random Forest, and AdaBoost with the features: $\{F_1$-$F_8\}$.
Sruti Bhagavatual *et al.* (Bhagavatula et al., 2014) in 2014 designed a machine learning based approach to classify ad-URLs using the AdBlockPlus classification system. The features they used in their study can be categorized into lexical-based feature and URL-based feature. Their evaluation on the classification models was conducted by the classifiers such as NB, SVM, LR and KNN, along with the features: $\{F_9$-$F_{15}\}$.
Pradeepthi.K and Kannan.A (Pradeepthi and Kannan, 2014) in 2014 provided a solution for the problem of phishing URL detection based on machine learning techniques. They concluded that the tree-based classifiers showed better performance for identifying phishing URLs. In their paper, totally four types of features were selected for the purpose of classification: lexical-based features $\{F_9, F_{17}, F_{26}, F_{27}\}$, URL-based features $\{F_{10}, F_{28}\}$, network-based features $\{F_{29}, F_{30}\}$, and host-based feature $\{F_{19}\}$.
In summary, machine learning techniques have been extensively used to detect suspicious URLs. A total of 36 features are mentioned in the previous papers, which can be grouped into 4 main types: Textual-based features, URL-based features, Host-based features and Network-based features (See in Table 1).

3 FEATURES

A growing number of studies focus on URL-based webpage identification. Beyond rule-based techniques such as blacklist or whitelist, the application of URL-based classification is significant for identifying and detecting malicious advertisement. Since it can protect end users from involving any malicious and fraudulent activities before downloading the webpage content. Also, the URL-based classification system can be more stable to the evolution of obfuscation or choaking techniques with less human involvement.
Among all the features we mentioned in Table 1, lexical-based features are the most readily available ones. Lexical-based features can provide sufficient information about the webpage content. They have been used in previous researches in (Szczepański et al., 2013) (Kan and Thi, 2005) (Ma et al., 2011) (Baykan et al., 2011) (Baykan et al., 2009) (Devi et al., 2007) (Zhang et al., 2006) (Bhagavatula et al., 2014) (Pradeepthi and Kannan, 2014) (Chen et al., 2014) (Le et al., 2011). Compared with other types of features, such as host-based features or network-based features, using lexical features alone can also

lead to a comparable classification accuracy (Le et al., 2011) without any time latency issues or resource consuming issues. The high accuracy and the lightweight properties make lexical-based features potential candidates for classifying malicious advertisements.

3.1 Available Features

We test and evaluate the performance of the following 7 features that have been previously used to identify online advertisements.
Textual Length of URL: The length of a URL without considering 'www.' .
Textual Length of Hostname: The length of hostname in a given URL.
Numeric Tokens Count: A binary value indicating if there is any numeric tokens in a given URL.
Ad-related Keywords: A binary value indicating if there is any ad-related keywords in a given URL, such as "ad, advert, popup, banner, sponsor, iframe, googlead, adsys, adser" .
Suspicious Symbol: A binary value indicating if there is any suspicious symbol in a given URL, such as " @ ; ".
Dots Occurrences: The number of dots in given URL.
Dash Count in Hostname: The number of dash occurrences in the hostname of a given URL.

3.2 New Proposed Features

URL obfuscation techniques are commonly used in phishing and malicious attacks (Ma et al., 2009) (Le et al., 2011) (Lin et al., 2013). For example, the attackers can obfuscate the host with an IP address, large host name or another domain. In order to identify phishing URLs, several lexical-based features are used in (Le et al., 2011) to address different types of obfuscation techniques: (I) **Features related to domain name**. These features include the length of the domain name, the length of the longest token in domain. (II) **Features related to filename or path**. These features include the length of the filename or path, the length of the longest token in filename or path. (III) **Features related to delimiters**. These features include the number of dots and delimiters (such as '-'). In addition, some other lexical-based features are used to detect malicious URLs (Mamun et al., 2016) (Lin et al., 2013). For instance, length ratio of different components in URL can help to find the abnormal component.

Although the features mentioned above can provide extra information about the suspicious URLs, and are common candidates for detecting phishing or malicious URLs, they have never been used for advertisement URL identification. In order to characterize the obfuscation techniques in malicious advertisement and improve the accuracy of our system, the following 26 lexical-based features are used as new features in our experiment:
Length of Domain: The length of domain in a given URL.
Length of Filename: The length of filename in a given URL.
Longest Token Length: The length of longest token in a given URL.
Average Token Length: The average length of all the tokens in a given URL.
Longest Path Token Length: The length of the longest token in the path of a given URL.
Average Path Token Length: The average length of all the tokens presence in the path of each URL.
Number of Symbols: The number of symbols in a given URL, such as "() [] // - + = . / ? : ! ,".
URL Token Count: The number of tokens in a given URL.
Length Ratio: We check the length division of Domain name / URL, Path / URL, Reference / URL, Query / URL, Path / domain, Reference / domain, Query / domain, Reference / path and Reference / query as length ratio features.
URL Pattern-based Features: We want to check if there is any specific pattern for online advertisement URL. The following features are considered as URL pattern-based features.

- **Letter-digit-letter:** If there is a digit presents between two letters in the given URL.
- **Digit-letter-digit:** If there is a letter presents between two digits in the given URL.
- **Delimiter Count:** The number of delimiters in the given URL.
- **Letter Count:** The number of letters in the given URL.
- **Digit Count:** The number of digits in the given URL.
- **Continuity rate:** We categorize the character type in the URL as letter, digit and symbol. Then we record the longest length for each type. For example, the continuity rate for URL "*go0gle12*@*" is (3 + 2 + 2) / 10 = 0.7.
- **Number Rate:** The proportion of digits in the given URL.

Executable File: If the given URL contains ".exe".
IP as Domain: If the domain name is an IP address.
Finally 33 lexical-based features are selected and used in our paper.

Table 2: Summary of the dataset and the experiment.

Scen.	Data Source	Feature Set		Algorithm		
Scen. A	*Benign-ad (3000) vs Non-ad (5000)*	Selected Set	Full Set	C4.5	RF	KNN
Scen. B	*Malicious-ad (1115) vs Non-ad (5000)*	Selected Set	Full Set	C4.5	RF	KNN
Scen. C	*Ad-URL (4115) vs Non-ad URL (5000)*	Selected Set	Full Set	C4.5	RF	KNN

4 DATASET

4.1 Previous Datasets

Viktor Krammer (Krammer, 2008) chose the Alexa Global Top 500 list of popular websites in their study. A total of 502 pages were examined in their experiment and 314 (63%) of them displayed some forms of advertising.

Zhou Li *et al.*, (Li et al., 2012) continuously crawled the home pages of Alexa's top 90,000 websites from Jun 21st to Sep 30th, 2011. They found that 53,100 webpages are involved in ad-related delivery. They also discovered that over 1% of the top Alexa home pages lead to malicious advertising.

Justin Ma *et al.* (Ma et al., 2009) collected benign URLs and malicious URLs separately. For benign URLs, they choose two data source, DMOZ Open Directory Project (Netscape, 2007) and random URL selector for Yahoo's directory, with a total number of 15,000 benign sources. For the malicious sites, PhishTank (OpenDNS, 2007) (5,500 malicious sources) and Spamscatter (Anderson et al., 2007) (15,000 malicious sources) were selected in their research.

In a similar study , Justin Ma *et al.* (Ma et al., 2011) chose two dataset feeds in their experiment. Their malicious URLs came from a large web mail provider, who provided 6,000+ examples of spam and phishing URLs per day. And their benign URLs were randomly collected from Yahoo's directory listing. Overall a total of 20,000 URLs per day were used in their research, which included about 30% malicious sources.

Sruti Bhagavatula *et al.* (Bhagavatula et al., 2014) selected the Alexa top 500 US sites from February 2014 as their dataset input, and crawling all the links from a page up to depth 2 from the source page, which led to a total of 60,000 URLs with 50% ad-related sources.

Caitlin R. Orr *et al.* (Orr et al., 2012) used 339 websites from Alexa, which distributed in 16 categories and 24 countries. They selected 250 unique scripts from the 339 websites for training purpose. They tested their classifier on scripts from 25 randomly chosen websites from Alexa Top 100,000.

4.2 Our Dataset

The main contribution of this research is the characterization of online advertisements. We decide to build a dataset which includes normal URLs (Non-ad), Benign-ad URLs and Malicious-ad URLs. Upon inspecting all the datasets listed in the previous subsection, we found that none of them has included all these materials together. So, this section describes the dataset we have generated and used in our testing and training processes. Three types of URLs sources are collected in order to conduct the classification task.

Source I (Malicious-ad URLs): We obtained examples of malicious advertising URLs from a URL blacklist service website, *www.urlblacklist.com*. The data sources are all about malicious advert servers and banned URLs, the latest modification date of the dataset is July 28th, 2016.

Source II (Benign-ad URLs): We collected benign advertising URLs from an advertising dataset, *www.code.google.com/archive/p/open-advertising-dataset/*, which is created by the University College London as an independent computational advertising dataset from the publicly available sources.

Source III (Non-ad URLs): We built a crawler to fetch all the *non-advertising URLs* from Alexa Top 5000. For each Alexa domain (such as *google.com*), the crawler visits at most 20 pages which are originally linked with Alexa top page, from August 28th, 2016 to September 10th, 2016. After that, 5000 *non-advertising URLs* are selected manually from the above fetched URLs.

Three scenarios are defined for the experiments based on these three types of URLs sources.

Scenario A: consists of pairing benign-ad URLs from Source II and non-ad URLs from Source III for the purpose of "detecting benign advertisements";

Scenario B: includes malicious-ad URLs from Source I and non-ad URLs from Source III for the purpose of "detecting malicious advertisements";

Scenario C: contains benign-ad URLs from Source II, malicious-ad URLs from Source I and non-ad URLs from Source III, in order to examine the task of distinguishing advertisements from normal URLs.

Table 2 shows a breakdown of the number of URLs

related to each scenario.
Based on (Krammer, 2008), average two-third (nearly 0.67) of Alexa top websites display some forms of advertising. So in order to simulate the real-world case as realistic as possible, we set the advertising URLs *vs* non-ad URLs ratio to 0.6 in Scenario A, and add 5% noisy data in each scenario. The distribution of positive labeled data and the negative labeled data are still uneven in Scenarios B and C (see in Table 2). Also, by dividing our URL sources into three scenarios, we can evaluate how our feature set performs in different circumstances, and estimate the potential use of URL lexical-based features in more applications.

5 EXPERIMENTS

In this section, we will first discuss how to select the most influential features for online advertising detection, and then introduce the implementation of our experiment.

5.1 Features Selection

Features may be considered as noisy features if they do not contribute to the detection performance. Instead of improving the performance of classifiers, the presence of such noisy features will be harmful, resulting in an increase in training time and error rates.

Based on different evaluation approaches, feature selection heuristics can separate the useful features from the unimportant ones. The selected features can deepen our knowledge about the nature of different data sources.
The estimations of feature selection algorithms depend on either a single feature or a subset of features (Szczepański et al., 2013). For example, the popular information gain heuristics can evaluate each single feature individually, but it cannot estimate the correlations between features. In order to measure the effectiveness of the features in our detection system, we used "*CfsSubsetEval*" in Weka toolbox, with "*BestSearch*" method. By considering the predictive ability of each feature, this evaluator can return a subset of features, which are highly correlated with the classification but having low inter-relation with each other (Xu et al., 2013) (Hall et al., 2009). The subset of features obtained from this approach is called "Selected Feature Set" and will be used in the following experiments. All the selected features in each scenario are list in Table 3.

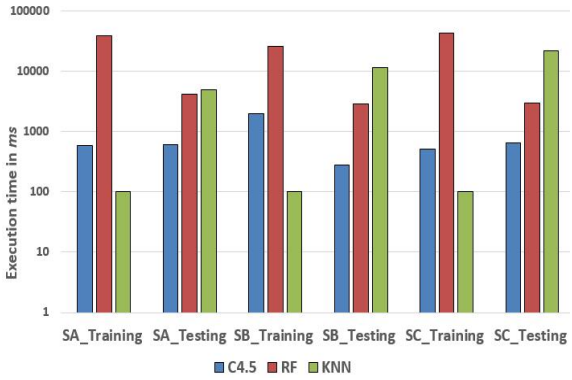

Figure 1: The execution time (in *ms*) of different classifiers in each scenario.

5.2 Classification and Evaluation

In our experiment, three different machine learning classifiers namely Decision tree algorithm (C4.5), Random Forest (RF) and K-Nearest Neighbor (KNN) have been used. the three algorithms are trained and tested on the Selected Feature Set (see in Table 3) and the Full-feature Set. In each scenario, we performed 10-fold cross validation to evaluate the results. The summary of different scenarios in our experiment is given in Table 2.

The Accuracy(Acc), F-score, and False Positive Rate (FPR) are used as evaluation metrics to assess the performance of the system, and are given by the following formulas:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$F-score = \frac{2*Pr*Rc}{Pr+Rc} \quad (2)$$

$$FPR = \frac{FP}{FP+TN} \quad (3)$$

Where TP, TN, FP, FN are true positive (detected ad-URL), true negative (detected legitimate URLs), false positive (undetected ad-URLs) and false negative (misclassified legitimate URLs). Pr and Rc represent precision and recall, and are given by formulas TP / (TP + FP) and TP / (TP + FN) respectively.
Table 4 shows the detailed results of our experiment. We notice that using the full feature set, we can always achieve high precision/recall and low FPR. This indicates that using lexical-based features alone can lead to a satisfactory classification effectiveness. The performance of selected feature set is still acceptable (see in Table 4). This shows that by reducing the dimension of full feature set with feature selector,

Table 3: Summary of selected features in each scenario.

Scen.	Selected Feature Set (Subset with BestSearch)
A	Ad-related keyword, Dash count, Average token length, URL token count, Domain / URL, Ref / URL, Path / domain, Number rate
B	Ad-related keyword, Suspicious symbol, Dash count, Length of filename, Path / URL, Number rate, Executable file, IP as domain
C	Ad-related keyword, Suspicious symbol, Dash count, URL token count, Domain / URL, Number rate, Executable file

Table 4: Summary of the experiment results.

		Full Feature Set			Selected Feature Set		
Scen.	**Algo.**	**Acc**	**F-score**	**FPR**	**Acc**	**F-score**	**FPR**
A	**C4.5**	97.6%	96.8%	1.5%	97.2%	96.8%	1.6%
	RF	97.3%	96.1%	1.7%	97.0%	96.4%	1.9%
	KNN	96.7%	95.7%	2.1%	95.9%	94.4%	3.7%
B	**C4.5**	97.5%	97.4%	0.6%	97.4%	93.4%	0.6%
	RF	97.4%	96.2%	0.9%	97.3%	96.1%	1.3%
	KNN	97.5%	96.5%	0.6%	96.5%	95.7%	1.8%
C	**C4.5**	97.4%	97.1%	2.2%	97.3%	97.0%	2.3%
	RF	97.1%	96.8%	2.2%	97.0%	96.8%	2.6%
	KNN	96.5%	96.1%	2.7%	93.1%	95.0%	4.4%

the selected features contain sufficient information for online ad-URLs, and are powerful attributes for online advertisement detection task.

Previously, many researches are working on URL-based webpages classification and detection. (Ma et al., 2009), (Choi et al., 2011), (Xu et al., 2013) and (Li et al., 2013) were focusing on malicious URL detection. (Le et al., 2011) and (Whittaker et al., 2010) designed automated classification systems for phishing URLs identification. (Szczepański et al., 2013) identified advertisements URLs by lexical-based analysis, whereas (Li et al., 2012) described a topology-based framework for malicious advertisement detection. However, our paper is the first study to deploy a comprehensive system for detecting benign-ad URLs and malicious-ad URLs at the same time.

A lightweight detection system is significant for online advertisement detection task, since any delay of classification can cause unexpected results for customers. Although features like WHOIS properties contain useful information about the URL instances, and are widely used in previous researches such as (Ma et al., 2009) (Ma et al., 2011) (Pradeepthi and Kannan, 2014) and (Li et al., 2012), they will cause time latency issues by sending requests to remote servers. And these issus will become more severe when the size of data is large. (Szczepański et al., 2013) demonstrated a classification system for benign-ad URLs with lexical-based analysis. However, since the introduction of "n-gram" and "bag of word" techniques, the size of their lexical-based features is large. This leads to a problem that the training time for their classification system is long, and the time delay will still suffer user experience. According to (Whittaker et al., 2010), online blacklist system will averagely take 1 hour to 10 hours to identify a single malicious URL. Based on Figure 1, the maximum execution time is only *40s* (using Random Forest algorithm in scenario A), which indicates that our lightweight detection system shows a considerable improvement over the existing blacklist approaches.

In our study, a lexical-based feature set with size 33 is selected and used in our experiment. Compared with previous works, our study proposes a lightweight solution for online advertisement detection. By using only lexical-based features, we can achieve a high classification performance and avoid the overhead of querying remote servers.

6 CONCLUSIONS

Nowadays, online advertisement is one of the largest annoyances for users in web surfing and reading. Current tools are using a large list of regular expression filters for matching every requested URL to detect the malicious-ads. So it is necessary to have an automated solution that can be adapted to thousands evolving adverts easily and fast. To tackle this issue, in this research we design and implement a lightweight classification system using lexical-based features. For the experiment, three different scenarios have been de-

fined based on three different sources: non-ad URLs, benign-ad URLs, and malicious-ad URLs. The results show that by using the selected lexical-based features, online advertisement detection accuracy is about 97% in certain scenario.

REFERENCES

Adobe (2013). The State of Online Advertising. https://www.adobe.com/aboutadobe/pressroom/pdfs/Adobe_State_of_Online_Advertising.pdf. (Accessed date September 2016).

Anderson, D. S., Fleizach, C., Savage, S., and Voelker, G. M. (2007). Spamscatter: Characterizing internet scam hosting infrastructure. *Usenix Security*, pages 1–14.

Andriatsimandefitra, R. and Tong, V. V. T. (2014). Capturing android malware behaviour using system flow graph. In *International Conference on Network and System Security*, pages 534–541. Springer.

Baykan, E., Henzinger, M., and Marian, L. (2009). Purely url-based topic classification. In *Proceedings of the 18th international conference on World Wide Web*, pages 1109–1110. ACM.

Baykan, E., Henzinger, M., Marian, L., and Weber, I. (2011). A comprehensive study of features and algorithms for url-based topic classification. *ACM Transactions on the Web (TWEB)*, 5(3):15.

Bhagavatula, S., Dunn, C., Kanich, C., Gupta, M., and Ziebart, B. (2014). Leveraging machine learning to improve unwanted resource filtering. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, pages 95–102. ACM.

Chen, C.-M., Guan, D., and Su, Q.-K. (2014). Feature set identification for detecting suspicious urls using bayesian classification in social networks. *Information Sciences*, 289:133–147.

Choi, H., Zhu, B. B., and Lee, H. (2011). Detecting malicious web links and identifying their attack types. *WebApps*, 11:11–11.

Devi, M. I., Rajaram, D. R., and Selvakuberan, K. (2007). Machine learning techniques for automated web page classification using url features. In *International Conference on Computational Intelligence and Multimedia Applications, 2007*, volume 2, pages 116–120. IEEE.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hernández, I., Rivero, C. R., and Ruiz (2014). Cala: an unsupervised url-based web page classification system. *Knowledge-Based Systems*, 57:168–180.

Hernández, I., Rivero, C. R., Ruiz, D., and Corchuelo, R. (2016). Cala: Classifying links automatically based on their url. *Journal of Systems and Software*, 115:130–143.

Jover, R. P., Murynets, I., and Bickford, J. (2015). Detecting malicious activity on smartphones using sensor measurements. In *International Conference on Network and System Security*, pages 475–487. Springer.

Kan, M.-Y. and Thi, H. O. N. (2005). Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326. ACM.

Krammer, V. (2008). An effective defense against intrusive web advertising. In *Sixth Annual Conference on Privacy, Security and Trust, 2008*, pages 3–14. IEEE.

Le, A., Markopoulou, A., and Faloutsos, M. (2011). Phishdef: Url names say it all. In *INFOCOM, 2011 Proceedings IEEE*, pages 191–195. IEEE.

Li, Z., Alrwais, S., Xie, Y., Yu, F., and Wang, X. (2013). Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 112–126. IEEE.

Li, Z., Zhang, K., Xie, Y., Yu, F., and Wang, X. (2012). Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and Communications Security*, pages 674–686. ACM.

Lin, M.-S., Chiu, C.-Y., Lee, Y.-J., and Pao, H.-K. (2013). Malicious url filteringa big data application. In *big data, 2013 IEEE international conference on*, pages 589–596. IEEE.

Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1245–1254. ACM.

Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. (2011). Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):30.

Mamun, M. S. I., Rathore, M. A., Lashkari, A. H., Stakhanova, N., and Ghorbani, A. A. (2016). Detecting malicious urls using lexical analysis. In *International Conference on Network and System Security*, pages 467–482. Springer.

Netscape (2007). DMOZ Open Director Project. http://www.dmoz.org. (Accessed date September 2016).

OpenDNS (2007). PhishTank. http://www.phishtank.com. (Accessed date September 2016).

Orr, C. R., Chauhan, A., Gupta, M., Frisz, C. J., and Dunn, C. W. (2012). An approach for identifying javascript-loaded advertisements through static program analysis. In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, pages 1–12. ACM.

PageFair (2015). Pagefair and adobe 2015 ad blocking report. https://pagefair.com/blog/2015/ad-blocking-report/. (Accessed date September 2016).

Pradeepthi, K. and Kannan, A. (2014). Performance study of classification techniques for phishing url detection. In *2014 Sixth International Conference on Advanced Computing (ICoAC)*, pages 135–139. IEEE.

Shih, L. K. and Karger, D. R. (2004). Using urls and table layout for web classification tasks. In *Proceedings of the 13th international conference on World Wide Web*, pages 193–202. ACM.

Szczepański, P. L., Wiśniewski, A., and Gerszberg, T. (2013). An automated framework with application to study url based online advertisements detection. *Journal of Applied Mathematics, Statistics and Informatics*, 9(1):47–60.

Whittaker, C., Ryner, B., and Nazif, M. (2010). Large-scale automatic classification of phishing pages. In *NDSS*, volume 10.

Xu, L., Zhan, Z., Xu, S., and Ye, K. (2013). Cross-layer detection of malicious websites. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 141–152. ACM.

Yu, F. (2015). Malicious url detection algorithm based on bm pattern matching. *International Journal of Security and Its Applications*, 9(9):33–44.

Zhang, J., Qin, J., and Yan, Q. (2006). The role of urls in objectionable web content categorization. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 277–283. IEEE.