Hindawi
Security and Communication Networks
Volume 2020, Article ID 6662166, 14 pages
https://doi.org/10.1155/2020/6662166

Research Article

Detecting Web Spam Based on Novel Features from Web Page Source Code

Jiayong Liu,[1] Yu Su,[1] Shun Lv,[2] and Cheng Huang[1]

[1] *College of Cybersecurity, Sichuan University, Chengdu, China*
[2] *College of Computer Science, Sichuan University, Chengdu, Sichuan, China*

Correspondence should be addressed to Cheng Huang; opcodesec@gmail.com

Received 1 November 2020; Revised 24 November 2020; Accepted 4 December 2020; Published 17 December 2020

Academic Editor: Liguo Zhang

Search engine is critical in people's daily life because it determines the information quality people obtain through searching. Fierce competition for the ranking in search engines is not conducive to both users and search engines. Existing research mainly studies the content and links of websites. However, none of these techniques focused on semantic analysis of link and anchor text for detection. In this paper, we propose a web spam detection method by extracting novel feature sets from the homepage source code and choosing the random forest (RF) as the classifier. The novel feature sets are extracted from the homepage's links, hypertext markup language (HTML) structure, and semantic similarity of content. We conduct experiments on the WEBSPAM-UK2007 and UK-2011 dataset using a five-fold cross-validation method. Besides, we design three sets of experiments to evaluate the performance of the proposed method. The proposed method with novel feature sets is compared with different indicators and has better performance than other methods with a precision of 0.929 and a recall of 0.930. Experiment results show that the proposed model could effectively detect web spam.

1. Introduction

With the rapid development of the network, web applications are becoming more and more popular in the recent years, among which search engines are one of the most common web tools for people to gain information every day [1]. As the most popular search engine worldwide, Google processes over 40,000 search queries every second on average, which translates to over 3.5 billion searches per day and 1.2 trillion searches per year worldwide in 2012 [2]. There are data [3, 4] indicating that 85% of Internet users find websites through search engines and 90% of Internet users do not go past the first three pages on search results. Spammers design pages delicately to improve rankings as most users only access the first page of search results. There has been a brief definition of web spamming in the literature [5]; shortly speaking, web spamming is a black-hat search engine optimization (SEO) that deceive search engines to increase the ranking of a page in search engine results. These web pages are called web spam. As evident, spammers try to deceive search engines and attract end users to click on web spam sites. They not only reduce the effectiveness and efficiency of search engine results since web spam pages take much time to process but may also be full of malicious content and links. Lina et al. [6] present threats and related attacks about web spam. Although search engine companies have utilized various methods to counter spam [7], it is still a challenge to prevent the increase of black-hat SEO technology and the growth of spam pages nowadays. Therefore, it is of great significance to detect web spam with efficiency and accuracy.

Many researchers and experts have conducted much research on spam in this field. Several researchers have relied on feature extraction from text and links on the web page [8, 9]. Some other researchers detect web spam by crawlers observing different versions of the web page returned to search engines and ordinary users [10, 11], as well as there are methods based on spam purposes and user access logs [12]. Detection methods have also evolved continuously from the original statistical characteristics to determine whether a web page is spam to automatic monitoring using machine learning and deep learning, and the efficiency and accuracy of detection are also continuously improved. We are motivated by previous work in the field of web spam detection and cybersecurity, which has proved the viability of web spam detection using a combination of machine learning and effective features. We use similar insights to support the discovery of web spam based on novel features. Our method is different from that in previous studies in that we extract not only link-based statistical features but also semantic features based on text content analysis and structural features based on the structure of web pages from the source code. Additionally, in terms of real-world aerial applications, the proposed method could be deployed in the browser. For example, the judgment of each web page in the users' search results by the proposed method can provide constructive conclusions to users and browser manufacturers. As the proposed method has features based on semantics, it is helpful to detect spam in web pages where links to spam content are easily injected into.

In this paper, we proposed a method using machine learning algorithm RF that combines feature extraction and feature selection to classify whether a web page is spam or not. Note that, for a binary classification problem, the classifier aims to distinguish the web page as spam or nonspam. The main contributions of this paper are as follows:

(1) This paper considers some previously undescribed features for web spam detection. We extract three novel feature subsets by studying homepage's links, texts, and structure based on statistical and semantic similarity analysis. The experimental results prove that the importance of novel features ranks high and effective.

(2) This paper applies a feature selection method for precomputed features related to the homepage to reduce computational consumption and improve accuracy. We introduce random forest algorithm for building the web spam detection model. The method could automatically distinguish web spam and a normal page from the website homepage.

(3) We evaluate the proposed method with comprehensive evaluation metrics for binary classification problems. Our method achieves the F1 score of 92.9%, which is higher than that of the existing methods. The experimental results show that our method can effectively detect web spam.

The rest of this paper is organized as follows. Section 2 presents related work regarding web spam detection. Section 3 describes the proposed approach in detail, and Section 4 evaluates the proposed method and the results of our experiments. Finally, we discuss our conclusions and future work.

2. Related Work

Web spam is often categorized into four classes: content spam, link spam, cloaking, and redirection. Several researchers and experts present kinds of methods to combat web spam correspondingly.

There were many research methods from different perspectives in the early stages. For example, Jakub Piskorski et al. [13] explored linguistic features focused on the utility of content-based linguistic features with computing 208 linguistic attributes, Benczúr et al. [14] conducted commercial intent analysis because other than the ordinary methods depending on the website itself; the authors thought much web spam was for commercial purpose, Bíró et al. [15] applied an extension of latent Dirichlet allocation (LDA) which is a linked LDA technique for web spam classification since topics are propagated along with links in such a way that the linked document directly influences the words in the linking document, and Liu et al. [16] analyzed web spam with user behavior where user visiting patterns of spam pages and three user behavior features are proposed to separate web spam from ordinary ones. Luca et al. [17] studied the spectrum of black-hat cloaking techniques that target browser, network, or contextual cues to detect organic visitors. Their anticloaking system can detect whether a web page would split view content returned to two or more distinct browsing profiles.

With machine learning developing by, a plurality of popular machine learning algorithms combined with sorts of feature engineering methods are applied to detect web spam. Machine learning techniques are more flexible than other methods; some difficult problems can be solved and more accuracy becomes a reality. Liu et al. [18] used a sentiment analysis model based on topic enhanced word embedding to obtain more complete text context information. The document topic distribution matrix is used to extract the document features. Reza Mohammadi et al. [19] proposed a method to improve support vector machine (SVM) algorithm by using two nonlinear kernels in twin support vector machine (MKTWSVM), which was experimented on both UK-2006 and UK-2007 datasets. The authors used a language-model approach and qualified-link analysis on detection. Fdez-Glez et al. [20] proposed a new framework according to combine different techniques, particularly suitable for filtering spam content on web pages. Mei et al. [21] proposed an improved PageRank algorithm based on web page differentiation (DPR), which evaluates pages authority according to its links' numbers and assigns corresponding weights according to its authoritativeness when assigning PageRank values. They combined DPR with K-means, designed a differentiation page-based K-means algorithm. Jelodar et al. [22] presented a systematic framework based on the chi-squared automatic interaction detector algorithm and a modified string matching algorithm. The author used the modified knuth–morris–pratt algorithm to extract features from Alexa Top 500 Global Sites and Bing search engine results in 500 queries; then, they generated a tree model with useful attributes that can detect web spam. Asdaghi and Soleimani [23] proposed a new backward elimination feature selection approach with the Naive Bayes (NB) classifier.

Many experts also used different neural networks and deep learning algorithms to detect web spam. Renato Moraes et al. [24] presented a performance evaluation of different models of artificial neural networks used to automatically classify and filter real samples of web spam based on their contents. Li et al. [25] introduced the deep belief networks and combined with the synthetic minority oversampling technique (SMOTE) and denoising autoencoder (DAE) algorithm to improve the classification performance of web spam. In [26], the authors presented a framework called FS2RNN, a feature selection scheme using recurrent neural networks (RNNs), for the classification of spam nodes. In this framework, the dataset is preprocessed before applying RNNs in which principal component analysis (PCA) is used for dimension reduction on the dataset and recursive feature elimination (RFE) is used for feature selection. Belahcen et al. [27] addressed the web spam detection problem by using the graph neural network (GNN) architecture, which can act as a mixed transductive-inductive model that is able to classify pages by using both the explicit memory of the classes assigned to the training examples and the information stored in the network parameters.

In addition to detecting traditional e-mail spam and web spam, there are many scholars studying spam in social media called social spam, such as spam based on blogs, tweets, and YouTube videos. Fu et al. [28] presented a framework detecting spammers by measuring how careful a user is when she is about to follow a potential spammer. Samsudin et al. [29] proposed a framework that extracted features by using data collected from the YouTube spam dataset to detect YouTube comments spam. To deal with users who are affected by social spam, Ezpeleta et al. [30] focused on mood analysis and all content-based analysis techniques. Based on these heuristic research studies, we can apply to the problem that needs to be solved in this paper.

3. Proposed Method

In this section, we discuss the proposed method framework, give a comprehensive process of mining novel features, and determine classification algorithm training for the detection of the web spam model presented in this paper. The framework of the proposed method is depicted in Figure 1. The input is the web pages of sorts of websites. The output is a list of web pages with predicted classification scores, where a higher score indicates that the web page is more likely to be web spam. The proposed method is composed of 3 components: the prepossessing, features, and detection model. The number in brackets is the number of features. Next, we will describe the functionality and specific implementation methods of each component in detail.

3.1. Data Augmentation. We design our method based on the WEBSPAM-UK2007 dataset [31]. In the labeled samples given by the original dataset, the proportion of spam is only 6%. One of the main challenges we face is that the data are very imbalanced. There is no doubt that machine learning algorithms are data-driven approaches. It means that the performance of the model is highly related to data. Therefore, to augment the data, we extract more original data from the results of previous studies on this dataset. The summary of the augmented data method is to select the labeled data from the labeling results with high accuracy of the previous studies on the dataset, which are used as the labels of our data. Detailed information about data augmentation is given in Section 4.1.

3.2. WARC Parser. First of all, the dataset is structured, but the complex structured data cannot be directly applied because it contains unnecessary information. We need to process raw data. The original web pages' HTML documents in each host are arranged in sequence and stored in separate Web Archive (WARC) format proposed by the Internet Archive. The WARC format is an extension of the ARC File Format that has traditionally been used to store "web crawls" as sequences of content blocks harvested from the World Wide Web. Figure 2 shows a code snippet of a WARC file. We can see that each capture in a WARC file is preceded by a one-line header (line 1) that very briefly describes the harvested content and its length. Next to the one-line header, HTTP protocol response headers (from line 5 to line 16) are recorded, and then, multiple lines of HTML documents (from line 18 to line 43) are followed. We develop a WARC parser to separate the blocks into multiple individual HTML documents one by one and store them in different file folders according to what the domain extracted from the one-line header uniform resource locator (https://chato.cl/webspam/datasets/uk2007/contents/excerpt.txt) field the HTML document belongs to. Since the domain of each host is different, the folder name is the domain name.

3.3. Homepage Extraction and Check. A website contains at least one web page, and some websites are up to several hundred pages. The results obtained by users searching for keywords in search engines are just one web page for users, and it is challenging to get all the web pages of the website to which the current web page belongs. In other words, getting all the web pages is not easy, but getting the homepage is still relatively simple. Moreover, the homepage is the core of a website, covering the main content that a website wants to express to those who are visiting the website. For example, some companies, governments, and schools' websites will display related information about companies, governments, and schools such as history, main business, and contact information on the homepage. Statistics show that web spam pages are more inclined to improve their rankings in search engine results pages, especially homepages. Figure 3 shows that the percentage of a homepage with the largest PageRank value among all pages on the website of spam websites is higher than that of nonspam websites. It indicates that when spammers create a website, they intentionally make the homepage with the highest ranking. Some well-known algorithms for calculating page rankings include PageRank Page Score and TrustRank [32]. Therefore, it is very representative to check whether the homepage is spam. To some extent, the homepage can represent whether the entire website is spam.

The next step is to determine which HTML document is the homepage of a website. In the process of parsing HTML documents from WARC files, we have judged whether a web

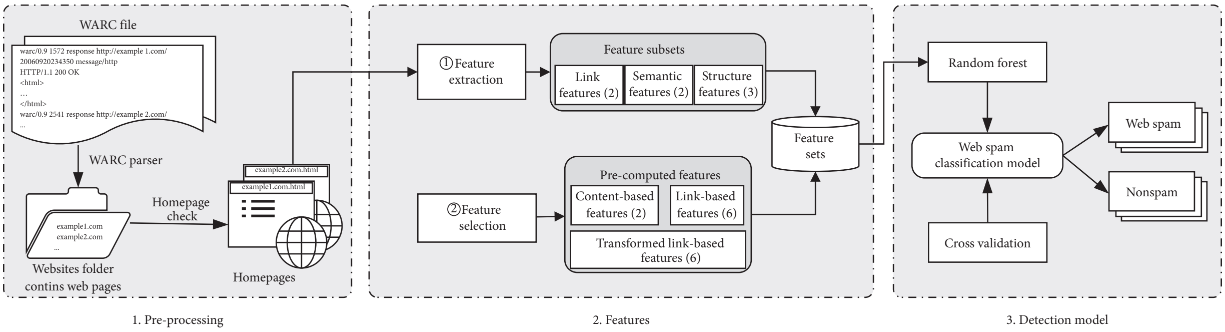


Figure 1: The framework of the proposed detecting web spam method.

```
warc/0.91572responsehttp://horwichrmicc.co.uk/20060920234350message/http uuid:e9004f91–fd17–4f40–b7d5–6c3e0b70f3d3
BUbiNG–guessed–charset: windows–1252
BUbiNG–content–digest: f05592c825fa9619306efcccd187391e

HTTP/1.1200 OK
x–powered–by: ASP.NET
connection: close
content–type: text/html
accept–ranges: bytes
content–location: http://horwichrmicc.co.uk/index.htm
server: Microsoft–IIS/6.0
content–length: 987
last–modified: Sun, 16 Apr2006 09:02:19 GMT
etag: “b5571d773461c61: b506b”
date: Wed, 26 Apr2006 13:11:10 GMT
ubi–http–equiv–charset: windows–1252

<html>
<head>
<meta http–equiv = “Content-Language”content = “en-us”>
<meta http–equiv = “Content-Type” content = “text/html; charset = windows–1252”>
<meta name = “horwich”“rmi”“winter hey lane”“pike”“reebok”“beehive”"rivington”“lostock”“blackrod”“lee lane”“chorley
new road” content = “Microsoft FrontPage 5.0”>
<metaname = “ProgId” content = “FrontPage.Editor.Document”>
<title>Horwich RMI CC</title>
</head>
<body>
<palign = “center”>
<ahref = “home%20.htm”><img border = “0” src = “bannerentry.gif ” width = “700” height = “400”></a>
</p>
<p align = “center”>
<b><script language = “JavaScript” type = “text/javascript” src = “http://pub24.bravenet.com/counter/code.php?id = 363671
&usernum = 2057536663&cpv = 2”>
</script><!-- End Bravenet.com Service Code--></b>
</p>
<p align = “center”>
<b><font size = “7”>Sponsored by Bespoke Flooring</font></b>
</p>
<palign = “center”>&nbsp;</p>
<p align = “center”>
<font color = “#FFFFFF”>THE OFFICIAL HORWICH R.M.I CRICKET WEBSITE</font>
</p>
</html>
```

Figure 2: An example data in the WARC format.

page is a homepage from the URL path roughly. We set every HTML document name as its pathname and store it under the website to which it belongs. Here are some simple rules, for example, the URL path is only the root path “/” could be as homepage, and the first-level path with the distinct keywords such as “index,” “home,” and “homepage,” is also the homepage. Of course, all web pages under some hosts do not match these rules, so a manual check is required.

3.4. Features. We extract some novel features from the source code of the web page and divide them into four categories mixed with existing features: homepage links features, semantic similarity features, homepage structure complexity features, and existing features. Although some features based on links, content, and structure have been used in previous papers, in this paper, we have studied these features from a different perspective.

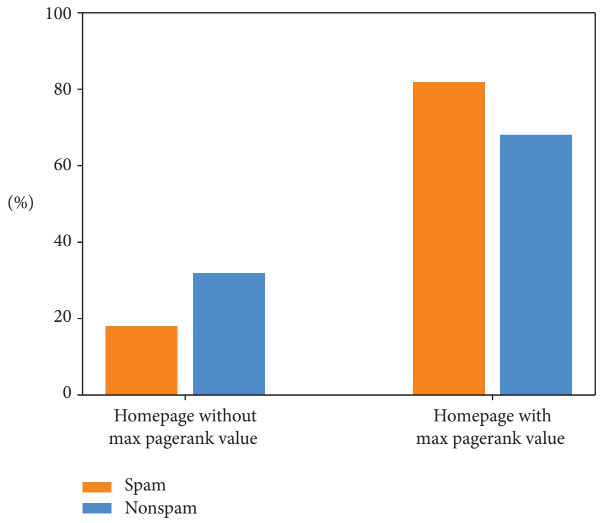


Figure 3: The percentage of homepage with max pagerank value of both spam and nonspam websites.

3.4.1. Link Features. It is necessary to consider link characteristics because the spammers deliberately set up a large number of hyperlinks between the spam websites to point to each other which can increase the clickthrough rate and the PageRank value of the website's homepage. There are some specific measures, such as inserting hyperlinks in the homepage to point to essential or well-known websites and attract other pages to point to their own pages or using information hiding technology to publish valuable information on the Internet, but hidden text or hyperlinks that are invisible to users, pointing to spam homepages. These practices all increase the entry link, also called indegree of the homepage of the spam websites. This makes the PageRank value of the homepage increase and leads to the advance of the spam page in the ranking of search engines, whereas the production of normal hosts is reasonable and standardized, and the host owner will not deliberately increase the homepage's incoming link. Many previous studies only focused on the number of all links, without considering external links and cross links separately. But, the impact of these two features on the construction of web spam is different. Based on this perspective, this paper extracts these two features separately. As discussed above, we propose two link features, number of external links and number of cross links, based on all the links in the homepage.

(1) Number of external links: external links defined as hyperlinks that point at an external domain which means any domain other than the domain the link exists on. We compared the domain of each link in the homepage with the domain to which the homepage belongs and counted the number of external links.

(2) Number of cross links: in contrast to external links, cross links also called internal links are links that, from within a website, point to another page which belongs to the same website. Similarly, the number of cross links is obtained by subtracting the number of external links from the total number of links.

3.4.2. Semantic Similarity Features. Generally, for content spam, the primary method is to repeat the same or similar keywords in large numbers. Some web pages directly copy the content of some standard high-quality websites. When users search for a specific keyword, these plagiarized websites will also have a relatively high ranking. Users will not be aware of this is spam page through only the restricted content displayed in the search engine results. These pages add partial anchor texts link to some marketing and authority websites, even malicious websites such as gambling and pornographic websites, to entice users to click and earn profits. This malicious behavior can be challenging to detect. Many web pages with interactive functions are easily used by spammers. For example, in the comments section of a blog, it is easy to evade censorship and spread malicious websites to entice users to click. In previous studies of content-based web spam, researchers mainly focused on the topics and keywords of the entire website. The method of detecting web spam from the entire content of the website is not accurate enough, and it will make web spam using this technology evade detection. Also, the partial web spam technique has not been widely studied yet. After analysis and manual check, we observe that the semantic analysis between the anchor text and the current web page is helpful for web spam detection. Therefore, we extract two semantic similarity features, namely, similarity of texts and links.

(1) Similarity of texts: the feature represents the semantic similarity between the textual description, also known as anchor text of the external links in the page and some textual description of the web page. It can reflect the similarity between the link's anchor text inserted in the web page and the main content of the web page.

(2) Similarity of links: the feature represents the semantic similarity of each domain of all links in the page and page's domain. It can reflect the similarity between the link inserted on this web page and the page's domain.

In this paper, we choose word mover's distance (WMD) as a metric to measure semantic similarity. The WMD is a novel distance function between text documents presented in [33]. In a supervised learning task, semantic similarities are useful for classification. WMD measures the difference between two texts and calculates the minimum distance that a word vector of one text "moves" to a word vector of another text. As shown in Figure 4, after removing stop words (not bold), the remaining words are embedded into a vector in the vector space. The WMD distance between the two short texts is the minimum cumulative distance calculated by word vectors in short text 1 travel to short text 2. Since the WMD distance can use the word-level semantic information represented by word2vec [34], it can achieve better results in the short text semantic distance calculation. Thus, WMD is suitable for computing the semantic similarity features in this paper. The smaller the WMD value, the more similar the two short texts. To automatically extract the semantic similarity features from the homepage's HTML

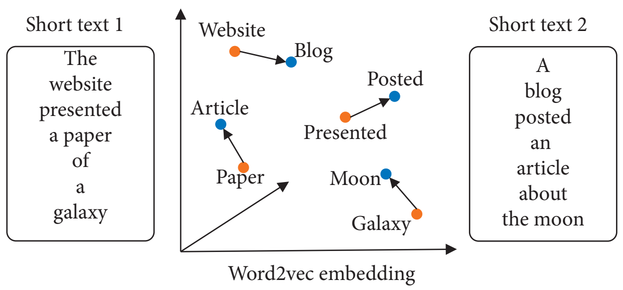


Figure 4: An illustration of the word mover's distance.

document, we propose an algorithm and complete it through the three steps: short text cleaning, represent words as vectors, and computing WMD distance. The pseudocode of this algorithm is shown in Algorithm 1.

As described in Algorithm 1, line 1 and line 2 show that the homepage and domain name of each website is required. Line 3 creates three lists: text_{hp} contains the homepage text, $\text{link}_{\text{external}}$ includes all external links, and $\text{text}_{\text{external}}$ is for storing the anchor text corresponding to each external link.

Line 4 to line 24 is the process of calculating semantic similarity features. There are five functions in the proposed method.

(i) **ExtractText()**: this function extracts the homepage's title, keywords, description in meta tags

(ii) **Collectlinks()**: this function extracts all external links of the homepage

(iii) **ExtractLinkText()**: it extracts the anchor text of each link in turn

(iv) **ExtractDomain()**: this function extracts the domain of each link in turn

(v) **WMD()**: it calculates the WMD distance between the homepage text and each anchor text of the external link and calculates the WMD distance between the homepage domain and each external link domain

Next, we introduce each step in detail.

Step 1 *Short Text Cleaning.* The HTML document of the homepage contains much information, but only a few parts are used to extract WMD features. To extract the title, the keywords and descriptions in the metafield, external links, and text description of every external link from the homepage of each website, we first need to parse the HTML tags with the Beautiful Soup Python library [35] and convert HTML entities to characters with the HTML Python library. Moreover, we remove some punctuations and stop words in raw texts. We utilize stop words from the NLTK library, which contains 127 English words. Besides, to ensure a hyperlink is an external link, it is necessary to extract each website's domain. Note that the external link includes neither relative paths nor links under the same domain which we mentioned in Section 3.1.1. Then, we splice the content of the homepage's title and keywords and description of metatag together as the text_{hp}. Besides, we should process two parts for each external link, the link itself and the anchor text of the link. Some websites contain more than one external link while some contain none. We push all the external links to a list as $\text{link}_{\text{external}}$. For each link's anchor text, we also push all the anchor texts corresponding to each link into the $\text{text}_{\text{external}}$ in turn, which is separated by spaces.

$$\begin{aligned}\text{text}_{\text{hp}} &= \left\{\text{text}_{\text{title}}, \text{text}_{\text{keywords}}, \text{text}_{\text{description}}\right\},\\ \text{link}_{\text{external}} &= \left\{\text{link}_1, \text{link}_2, \ldots, \text{link}_{\text{n}}\right\}, \quad n \in N^*,\\ \text{text}_{\text{external}} &= \left\{\text{text}_{\text{link}_1}, \text{text}_{\text{link}_2}, \ldots, \text{text}_{\text{link}_{\text{n}}}\right\}, \quad n \in N^*.\end{aligned} \tag{1}$$

Step 2 *Represent Words as Vectors.* Since models accept numerical input only and the words in short texts are natural languages such as English, these words need to be converted into numerical forms or embedded in mathematical space. The vector mapped to real numbers is called word vectors. The embedding method is called word embedding, and word2vec is a kind of word embedding method. Word2Vec is a tool to transform the text processing into a vector in the multidimensional vector space, representing the text's semantic similarity based on the similarity in the vector space. We use a pretrained model, Google News [36], to train word vectors. It contains 3 million pretrained English word embeddings.

Step 3 *Computing WMD Distance.* After obtaining the original short texts and links' word vectors, we then calculate the WMD distance to represent the similarity, which is illustrated as follows:

$$\text{WMD} = \min_{T\geq 0} \sum_{i,j=1}^{n} T_{ij} c(i, j). \tag{2}$$

T is a sparse matrix where T_{ij} is the weight of word i in document d_i move to word j in document d_j, $c(i, j)$ is the Euclidean distance between word i and word j, as equation (3) shows, and x_i, x_j is the word i, j's word vector after embedding, respectively.

$$c(i, j) = \left\| x_i - x_j \right\|_2. \tag{3}$$

The sum of weight of word i to another document $\sum_{j=1}^{n} T_{ij}$ is equal to the weight of word i in the first document d_i, and the sum of weight of word j to another document $\sum_{j=1}^{n} T_{ij}$ is equal to the weight of word j in the first document d_j.

$$\begin{aligned}\sum_{j=1}^{n} T_{ij} &= d_i, \quad \forall i \in \{1, \ldots, n\},\\ \sum_{i=1}^{n} T_{ij} &= d_j, \quad \forall j \in \{1, \ldots, n\}.\end{aligned} \tag{4}$$

d_i can be calculated by equation (5), where c_i is word i appear times in the document I.

$$d_i = \frac{c_i}{\sum_{j=1}^{n} c_j}. \tag{5}$$

In the case of this article, document I corresponds to the text_{hp} and document J corresponds to the links $\text{link}_{\text{external}}$ and anchor text $\text{text}_{\text{external}}$ of external links. So, there are multiple short texts. We need to compare the semantic similarity of each external link with the homepage. Then, we calculate the average by equations (6) and (7), respectively.

$$\text{WMD}_{\text{text}} = \frac{\sum_{i=1}^{n} \text{WMD}\left(\text{text}_{\text{hp}}, \text{text}_{\text{link}_i}\right)}{n}, \tag{6}$$

$$\text{WMD}_{\text{link}} = \frac{\sum_{i=1}^{n} \text{WMD}\left(\text{domain}_{\text{hp}}, \text{domain}_{\text{link}_i}\right)}{n}. \tag{7}$$

3.4.3. Structure Complexity Features. It is necessary to consider the characteristics of the homepage's document object model (DOM) structure because we discover that many web pages use the same templates, such as domain parking services, personal blog websites, and some government websites. It can be considered that there are certain regularities. According to MDN [37], a web page is a document of which structure and content are represented as nodes and objects by DOM. Structural features can reflect the complexity of web pages. Similar web pages have a similar DOM structure because web spam is often a well-designed template, unlike normal web pages that have different characteristics. The previous method of identifying web spam mainly focused on the difference in content and links, but we also consider the web page's structural characteristics in this paper. Previous methods also studied the structure of HTML, such as extracting the number of <a> tags and <img> tags. However, this method may not be very effective for some specific websites, such as online shopping websites and stock picture websites, which have an obvious tendency of certain types of HTML tags. We not only analyze a certain type of tags but also analyze the complexity of the web page's structure from a more general perspective. We have analyzed the three features of the number and diversity of HTML tags and the depth of element nodes. For example, the domain parking service, where the parking platform often has a fixed template. We aim to identify such a web page. Thus, it is necessary to research the web page's structure, and we have extracted the following structural features from the source code of the homepage.

(1) Number of HTML tags: the DOM represents an HTML document as a tree structure of tags, which is a DOM tree. We only consider the element-type nodes, HTML tags. By traversing the DOM, the total number of tags contained in the homepage is calculated.

(2) Diversity of HTML tags: different types of websites have different distributions of their tags. For example, the web pages in some link factories contain a large number of hyperlinks, and the ¡a¿ tag implements these hyperlinks. There are many image tags in online stores, while personal blogs have many paragraph tags. We also counted each type of label on each homepage.

(3) Depth of element nodes: by traversing each branch of the DOM tree, we calculated the maximum depth of the DOM tree of the homepage. It reflects the complexity of a DOM tree structure and the complexity of the homepage's HTML structure.

3.4.4. Precomputed Features. There are 277 precomputed features in total, which categorize into 4 sets, direct features set with 2 features, content-based features set with 96 features, link-based features set with 41 features and transformed link-based features set with 138 features, which are obtained by mathematically transforming the link-based features. If all features are considered for experiments, it is evident that those with high dimensions will undoubtedly consume a lot of resources and cause a long execution time. In fact, many features are redundant. By removing several redundant features, both efficiency and accuracy can be improved. Therefore, we only take the features related to the homepage into consideration.

First of all, by checking the meaning of each feature, we preliminarily filter out 106 precomputed features from these four feature sets that are all about the homepage (hp), without considering the page with the maximum PageRank (mp) value. Then we select features from the 106 features. We use a new backward elimination approach, Smart-BT, proposed by Asdaghi and Soleimani [23] to accomplish feature selection. This method differs from sequence backward elimination in that it measures the impact on the classification result after eliminating a set of features, rather than eliminating a single feature. In summary, we extracted 7 new features and selected 14 features from the existing features. There are 21 features in total. These features will be input into the detection model.

3.5. Classification Algorithm. Judging whether a web page is spam or not is a problem with less clear boundaries. It is a subjective issue to some extent, so classifying web pages as spam or nonspam is challenging. Because of the apparent differences between spam and nonspam web pages in some features, we can use these features to build machine learning models that allow experts or researchers to identify web spam and reduce losses on the ground quickly. Classic algorithms, such as NB, logistic regression (LR), SVM, RF, convolutional neural networks (CNNs), RNN, and long short-term memory (LSTM), have different advantages and disadvantages. In [38, 39], a large-scale empirical comparison between these machine learning methods is presented. A CNN has an excellent performance spatial mapping, such as image data. An RNN is more suitable for sequence content, such as text analysis. But, an RNN has the problem of gradient disappearance; it is challenging to process long

```
Require:
hp: homepage of each domain
$d_{hp}$: homepage's domain
Initialize the list $text_{hp}$, $link_{external}$ and $text_{external}$ set to null
if (hp is not null) then
    $text_{hp} := $ ExtractText (hp)
    /* extract hp's title, keywords, description */
    $link_{external} :=$ Collectlinks (hp)
    /* collect all external links in hp */
    if ($link_{external}$ and $text_{hp}$ is not null) then
        for each link $\in\in link_{external}$ do
            $text_{external} := $ ExtractLinkText (link)
            $d_{link} := $ ExtractDomain (link)
            /* extract link's anchor text */
        end for
        $WMD_{text} = $ WMD ($text_{hp}, text_{external}$)
        /* computing the WMD distance between hp's text and external link's anchor text*/
        $WMD_{link} = $ WMD ($d_{hp}$, $d_{link}$)
        /* computing the WMD distance between hp's domain and external link's domain*/
    else if ($link_{external}$ is not null and $text_{hp}$ is null) then
        $WMD_{text} = 0$
    else
        $WMD_{text} = WMD_{link} = 0$
    end if
end if
return $WMD_{text}$, $WMD_{link}$
```

Algorithm 1: Calculating semantic similarity features from the homepage.

sequence data. LSTM can avoid the vanishing of gradient of conventional as a special case of the RNN.

Considering the characteristics of the dataset is imbalanced and features in the feature set are independent and based on the cost of different methods, the RF [40], which combines a multitude of decision trees, is more suitable for the problem solved in this paper. As an ensemble learning method for classification, RF solves the shortcomings of the weak generalization ability of decision trees since it predicts a sample by lots of decision trees voting for the final result. Furthermore, there are several advantages to selecting RF as the classifier. It is inherently easy to interpret and understand. Furthermore, RF algorithm is easy to implement and costs less than deep learning. Therefore, we chose to use RF as the automatic classifier in this paper.

4. Experiments and Evaluation

In this section, we have first illustrated the environment of experiments and detailed the source and composition of the dataset used in this paper. Then, the metrics utilized for the measurement of the performance of the proposed model are discussed, and later, experiment results are analyzed.

The experiments studies are conducted on the Ubuntu operating system. The homepages extraction and data preprocessing are developed in some libraries written in Python. Also, the process of model building and classification is implemented by scikit-learn [41] and keras [42] with TensorFlow backend [43]. The experimental environment configuration is shown in Table 1.

Since this paper focuses on the extraction and effects of novel features, the parameters in the detection model should be set or adjusted as little as possible. The simpler the machine learning model, the more likely it is that good experimental results are not based solely on specific samples. For example, in the detection model RF, we only set the number of trees parameter "n_estimators" to be 100 empirically. In fact, the default value of "n_estimators" changed from 10 to 100 in scikit-learn v0.22. There is no specific setting for the other hyperparameters, which means other hyperparameters are set by default. The advantage is that the classifier of this paper is not aimed at a specific dataset, but has generalization capabilities. There is a reason to believe that the proposed method is not too data dependent and easy to apply for new users. When encountering a new dataset, we only need to extract the features proposed in this paper according to the method in Section 3 and input these features into the classifier for classification. However, machine learning is dependent on data, and different data types have different targeting models, which we mentioned in Section 3.2. For data with similar regularities, the proposed method has generalization ability. Firstly, different web spam pages have similar characteristics, such as too many links for link-based web spam or a large number of repetitions of text content for content-based web spam. Secondly, cross-validation is used to evaluate the prediction performance of the model, especially the performance of the trained model on new data. The cross-validation can reduce overfitting to a certain extent and better evaluate the generalization quality of the model by repeatedly dividing the dataset.

Table 1: Experimental environment Configuration.

Designation	Information
Operating system	Ubuntu server 16.04 LTS
System configuration	CPU:Intel i7-7700 K, 3.60 GHz, 32 GB RAM GPU:Nvidia RTX 2080 8G
Python library	Genism 3.8.1 TensorFlow 2.0.0 Beautifulsoup4 4.4.1 NLTK 3.4.5 Keras 2.3.1 Scikit-learn 0.21.3 Matplotlib 3.1.1 Statsmodels 0.9.0
R library	pROC 1.16.2

4.1. Dataset. We run our experiments on the WEBSPAM-UK2007 dataset [31] which is a large collection of 105,896,555 pages in 114,529 hosts based on a crawl of the "uk" domain that was conducted in May 2007. It is also used as the Web Spam Challenge 2008 dataset. Although the amount of the dataset is large, only few were labeled by a group of volunteers. As shown in Table 2, among the all 6479 labeled data, we discard the data labeled as "undecided" because it means that the volunteers were still uncertain as to whether they were spam or nonspam and discard the data without content features. Meanwhile, we delete these data that their features are not complete. As a result, 5797 data remain.

As Table 2 depicts, the number of spam is 321, and the proportion of spam is only 6%; the ratio of samples of nonspam class to spam class is nearly 16 : 1, which means that the data are very imbalanced. In such scenarios, machine learning models cannot learn the characteristic behavior of the minority spam class. Classifying samples as spam or nonspam accurately presents considerable challenges. To address this issue, we re-extracted 1215 pieces of data after removing duplicate data in the original dataset from the results given by the top three [44] in the Web Spam Challenge 2008. These data were consistently labeled as "spam" by the top three teams. To a certain extent, these data can be considered reliable. Although we extracted more labeled data, these data were already 13 years old, so we also considered the newer dataset UK-2011 [45], which was derived from the WEBSPAM-UK2007 dataset. After deduplication, we find that all pages on many websites are invalid websites. "Invalid" means the source code of these pages has no content or is meaningless. There are probably the situations "301 Moved Permanently," "Object Moved," "This IP has been banned," and "302 Found." Since these pages have no research value, they need to be deleted. In addition, we have also removed some pages that are not in English. In the end, as Table 2 depicts, we have 6189 pages consisting of 4745 nonspam pages and 1444 spam pages.

We have a reason to believe that conducting the study with the 13-year-old dataset has certain limitations, such as whether it is suitable for today's rapid development of web applications. The meaningfulness of using the dataset is as follows.

Table 2: Statistics of experiments dataset.

Category	UK2007	UK2011	Dataset used in this paper
Nonspam	5476	1768	4745
Spam	321	1998	1444
Total	5797	3766	6189

(1) This dataset is a standard dataset in the field of web spam research, and its labels are judged by multiple scholars. It can be considered that its labels are authoritative.

(2) Although it has passed 13 years, it is still in the web 2.0 era now. Developers build web pages, especially web spam, with some commonality technologies before and nowadays, which indicates the dataset is universal. Moreover, during our manual check process, we find that some websites are still active.

4.2. Experimental Design. We conduct three sets of experimental studies to evaluate the performance of our model fully: (1) we first evaluate the performance of our proposed method and verify the effectiveness of the novel feature; (2) we also compare the performance of our method with some popular web spam detection systems; and (3) we use hypothesis testing to verify the validity of our method and analyze the importance of features.

4.2.1. Experiment for Performance of Proposed Model. We first examine the performance of the RF model on the dataset and compare the results with benchmark models. Considering the dataset is inadequate, especially the data for the minority class and different samples or different partitions of the dataset may cause the result to be optimistically biased, we take cross validation to train our dataset. As a potent tool in machine learning and deep learning, cross validation can ensure that every page in the dataset can be used in the experiment process. In this way, we ensure the full use of the data and manage to make the experimental results less biased. Thus, we apply 5-fold cross validation to train our dataset in all detection models. We input all the features that comprise existing features and novel features into classification approaches to determine whether a page is spam or not. We have also investigated some benchmark traditional machine learning algorithms such as NB, LR, and SVM and some benchmark deep learning algorithms including CNN, RNN, and LSTM as basic comparative experiments. Secondly, we compare the classification effects of each model on only existing features and all features.

4.2.2. Experiment for Comparison of Detection Rate. We also have compared some state-of-art methods; for example, Mittal and Juneja [46] presented a mutual information-based feature selection method which selects content-based features and with a SVM classifier to distinguish web spam, Makkar et al. [26] used PCA and RFE to deal with link-based features and incorporated the features into an RNN classifier to detect spam, and Asdaghi and Soleimani [23] proposed an effective feature selection method to select fewer features and put the features into NB model to achieve high performance. We used the same method as these papers to divide the dataset into training, validation, and test sets. In order to make a comparison on the dataset used in this paper, we reproduced these experiments.

4.2.3. Experiment for Validity of the Proposed Features. As one of the models considered is the LR model and the best model is RF, followed by comes LR, we believe that reporting the logistic regression model results with statistical inference would be very useful for more than one reason. We use statsmodels [47] for the estimation of many different statistical models. Firstly, it would verify that some of the features identified in Section 3.1 can actually be used to detect spam, and it would demonstrate which variables are the most important in this regard. Secondly, it could then be used as a benchmark against which the other models could be compared. We use the open-source package "pROC" provided in Robin et al. [48] to compare two different models' area under the curve (AUC). It helps us compare the superiority of different models more rigorously, especially when the *p* value is less than 0.05, and the results are more convincing.

We also use two methods including mean decrease impurity (MDI) and mean decrease accuracy (MDA) also known as permutation importance (PI) in RF for feature importance analysis. There are two main problems of impurity-based feature importance methods are that it biased towards high cardinality features, and the impurity-based importances are computed on the training set. Hence, it is not certain that features are also useful on the test set, whereas MDA is an alternative that can mitigate those limitations. We add up the ranked results of each feature and calculate the average importance score because of cross validation.

4.3. Evaluation Metrics. In binary classification problems, the most popular performance evaluation indicators are accuracy, precision, recall, and F1 score, which are described in detail as follows.

Accuracy is the number of correct predictions over the number of total predictions of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{8}$$

We judge a sample with a prediction score greater than the threshold as positive (spam class), and the threshold is 0.5. Where true positive (TP) is the number of spam samples that are correctly classified, true negative (TN) is the number of nonspam samples that are correctly identified, false negative (FN) is the number of spam samples that are mistakenly classified as a nonspam sample, and false positive (FP) is the number of nonspam samples that are mistakenly classified as spam.

Recall, also called true positive rate (TPR), is the proportion of the number of spam samples was identified correctly as spam in all spam samples and defined in equation (9). Precision is the proportion of the true predictions of the spam samples over total samples predicted as spam which is defined in equation (10).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{10}$$

F1 score is the harmonic average of precision and recall and defined in the following equation:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

False positive rate (FPR) is defined as follows:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{12}$$

Receiver operating characteristics (ROC) curve must be regarded as a widely used indicator when it comes to metrics for performance evaluation in classification problems. Because the ROC curve has an outstanding characteristic, the ROC curve can remain unchanged when the distribution of positive and negative samples in the test set changes. Also, AUC, which utilizes TPR and FPR (equations (9) and (12), respectively), represents classifiers' performance. The larger the AUC value, the better the classifier is in detecting web spam.

4.4. Results and Analysis

4.4.1. Classifier Performance Result. We show the performance of the binary classifiers in detecting web spam in Table 3, which demonstrates the results of different baseline models using the evaluation indicators described in Section 4.3. Figure 5 illustrates the graphical view of the performance of the different classifiers in web spam detection. It can be seen from the table and figure that the RF model yields the best performance in all aspects of our experiments, with a precision rate of 0.929 and a recall rate of 0.930 and has the largest curve area. Trees of RF algorithm are independent during the training process. The final result is obtained by voting of all trees. For imbalanced data sets, RF can balance errors [49]. LR is closely followed, and SVM, CNN, and RNN perform well but relatively worse. However, NB and LSTM performance are slightly worse. NB is relatively simple, more sensitive to minority class data. LSTM is suitable for longer sequence data, so the dataset in this paper does not highlight its advantages. The result of experiment for performance of the proposed model demonstrates that the RF model can use the features effectively. We can conclude that the selected novel features combined with the chosen classifier RF yields better results. In Table 4, we can see the existing features and novel features, as well as the results of all features under different evaluation indicators. Without novel features, the result is inferior to that of with novel features, which means that the novel features we extracted are practical.

Table 3: Results of different classification models. These bold values indicate the best performance under different evaluation metrics.

Model	Accuracy	Precision	Recall	F1 score
NB	0.850	0.843	0.850	0.844
LR	0.888	0.884	0.888	0.883
SVM	0.891	0.899	0.891	0.880
RF	**0.930**	**0.929**	**0.930**	**0.929**
LSTM	0.810	0.835	0.810	0.758
CNN	0.859	0.861	0.859	0.842
RNN	0.875	0.872	0.875	0.865

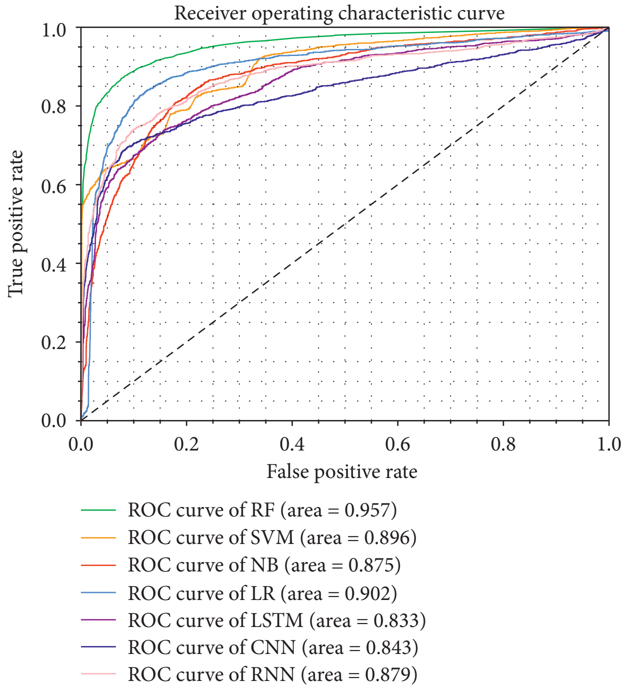


Figure 5: The ROC curve of 7 models (5-fold cross validation).

Table 4: Results of using different 3 feature sets on random forest model.

Feature set	Accuracy	Precision	Recall	F1 score	AUC
Selected existing features	0.911	0.909	0.911	0.909	0.937
Novel features	0.911	0.910	0.911	0.907	0.917
Selected existing + novel features	0.930	0.929	0.930	0.829	0.957

4.4.2. Comparison Experiment Result. In this paper, we reproduced three representative methods as comparative experiments that we explained in Section 4.2. The three studies were chosen for the comparisons based on the consideration that the researches were relatively new and their results performed well. Also, they all used the same dataset as this paper. As in Table 5, which demonstrates the results of three state-of-art methods, paper [46] achieved an F1 score of 0.883 on our dataset, which was less than our method by nearly 5%. Our method uses fewer features and achieves better results. It is concluded that these methods are worse than our method and our proposed method performs well.

4.4.3. Validity Verification Result. Table 6 demonstrates some regression results of the features used in this paper. It includes each feature's coefficients, standard error, and p value by logit regression analysis. It can be seen that the novel features contribute significantly to the model.

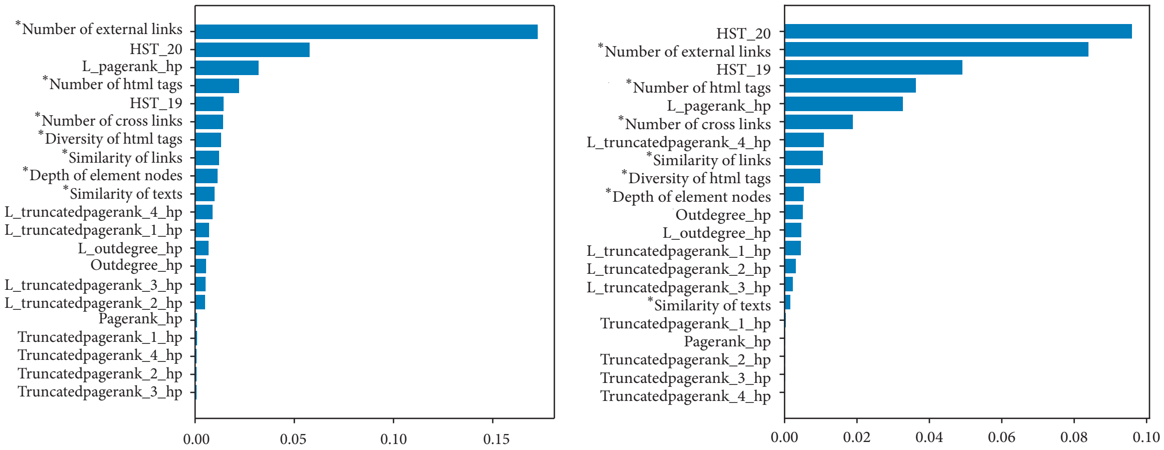
"Two ROC curves are 'paired' if they derive from multiple measurements on the same sample" described by Robin et al. [48]. Thus, we compare the ROC curve of RF with other models' ROC curves, respectively. We use "roc.test()" command from "pROC" package in R to calculate the p value. All paired ROC curves' p value is less than 2.2e-16, which is far less than 0.05. We could say that the RF model (AUC = 0.957) has an AUC that is significantly

Table 5: Comparison of the results using different classification methods.

Method	Number of features	Accuracy	Precision	Recall	F1 score
Our method	21	**0.930**	**0.929**	**0.930**	**0.929**
The method of Mittal et al. [46]	70	0.890	0.888	0.890	0.883
The method of Makkar et al. [26]	41	0.872	0.869	0.872	0.870
The method of Asdaghi et al. [23]	28	0.863	0.869	0.863	0.865

Table 6: Logit regression results.

Type	Feature	Coefficients	Std. error	Pr(>—z—)
Selected existing features	HST_19	0.7254	0.338	0.032
	HST_20	2.3106	0.420	0.000
	outdegree_hp	0.0066	0.003	0.015
	pagerank_hp	−9.712e + 04	4.2e + 05	0.817
	truncatedpagerank_1_hp	−1.043e + 05	1.35e + 06	0.939
	truncatedpagerank_2_hp	−1.079e + 05	2.35e + 06	0.963
	truncatedpagerank_3_hp	−1.101e + 05	4.28e + 06	0.980
	truncatedpagerank_4_hp	−1.084e + 05	2.74e + 06	0.968
	L_outdegree_hp	−0.0425	0.002	0.000
	L_pagerank_hp	0.0472	0.012	0.000
	L_truncatedpagerank_1_hp	6.4183	0.665	0.000
	L_truncatedpagerank_2_hp	−8.9288	1.766	0.000
	L_truncatedpagerank_3_hp	3.5543	2.380	0.135
	L_truncatedpagerank_4_hp	−0.9848	1.369	0.472
Novel features	Diversity of HTML tags	−0.0523	0.011	0.000
	Depth of element nodes	−0.0123	0.002	0.000
	Number of HTML tags	−0.043	0.000	0.000
	Number of external links	0.0460	0.003	0.000
	Number of cross links	0.0173	0.002	0.000
	Similarity of texts	−0.0816	0.029	0.004
	Similarity of links	1.7910	0.249	0.000



(a)

(b)

Figure 6: Random forest feature importance. (a) Mean decrease impurity. (b) Mean decrease accuracy.

greater than the second-best model (AUC = 0.902). Also, the results are not accidental which proves that our method is correct and effective.

As can be observed in Figure 6, the Figure 6(a) is the result of using MDI, and the Figure 6(b) is the result of using MDA. The results of the two RF feature importance ranking methods are not exactly the same. The ones with * on the *Y*-axis are novel features. It is clear that the novel features extracted in this paper rank top overall. The advantage of the features proposed in this paper is that it is convenient to extract, whereas the precomputed existing features extraction requires more stringent conditions such as construction of web graph. The novel features are general and easily accessible.

5. Conclusions and Discussion

Based on current research, this paper proposes a new method to distinguish web spam. We introduce a set of novel features about the homepage which we manually checked. In the meantime, we use the feature selection algorithm Smart-BT [23] to reduce the precomputed existing features' dimension so that the method's computational cost will decrease. Then, we use the RF model to discriminate against web spam with efficient identification. The experiment results showed that this method could reach a state-of-art level compared with other methods. Besides, the model with novel features which are are impressive to web spam detection is more superior and valid than the model with only existing features. Since this paper takes homepage only into account, the method is general and extensible because obtaining all pages of a website is not easy in most times. We acknowledge that some of the biases of our dataset might affect the result. Our method may not work well as the web spam evolves because the boundary between spam and nonspam is likely to blur. Also, we only analyzed statically from the source code without considering the dynamic parts such as JavaScript code, so our method has limitations for web spam that uses dynamic technology. For example, cloaking and redirection web spam. The proposed method only focuses on the homepage of a certain website without confirming whether the website returns different content for users and search engines so that there is a certain error in detecting this type of web spam. Moreover, many malicious websites redirect to other pages to improve rankings. There are many ways to achieve redirection, such as the redirection field in the meta tag and dynamic scripts in JavaScript. The proposed method does not pay attention to the JavaScript code and redirection web spam detection is not accurate enough.

In the future, mining more efficient features based on static and dynamic analysis and using a classifier with the ability of high accuracy would be an interesting direction. This will be the direction we will consider later.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no.61902265, Sichuan Science and Technology Program under Grant nos.2020YFG0047 and 2020YFG0076, and the Fundamental Research Funds for the Central Universities.

References

[1] Search engine marketing statistics 2020. https://www.smartinsights.com/search-engine-marketing/search-engine-statistics/.

[2] Google search statistics. https://www.internetlivestats.com/google-search-statistics/(2019).

[3] Google organic click-through rates in 2014, https://moz.com/blog/google-organic-click-through-rates-in-2014(2019).

[4] How far down the search engine results page will most people go? https://www.theleverageway.com/blog/how-far-down-the-search-engine-results-page-will-most-people-go/(2019).

[5] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," in *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan, May 2005.

[6] A. Lina, "Towards evaluating web spam threats and countermeasures," *International Journal of Advanced Computer Ence and Applications*, vol. 9, no. 10, 2018.

[7] M. Najork, *Web Spam Detection*, pp. 1–5, Springer, New York, NY, USA, 2017.

[8] M. Mahmoudi, A. Yari, and S. Khadivi, "Web spam detection based on discriminative content and link features," in *Proceedings of the 2010 5th International Symposium on Telecommunications*, Tehran, Iran, December 2011.

[9] R. K. Roul, S. R. Asthana, M. Shah, and D. Parikh, "Detecting spam web pages using content and link-based techniques," *Sadhana*, vol. 41, no. 2, pp. 193–202, 2016.

[10] J. Deng, H. Chen, and J. Sun, "Uncovering cloaking web pages with hybrid detection approaches," in *Proceedings of the 2013 International Symposium on Computational and Business Intelligence*, pp. 291–296, IEEE, New Delhi, India, August 2013.

[11] R. Duan, W. Wang, and W. Lee, "Cloaker catcher: a client-based cloaking detection system," 2017, https://arxiv.org/abs/1710.01387.

[12] Y. Liu, F. Chen, W. Kong et al., "Identifying web spam with the wisdom of the crowds," *ACM Transactions on the Web*, vol. 6, no. 1, pp. 1–30, 2012.

[13] J. Piskorski, M. Sydow, and D. Weiss, "Exploring linguistic features for web spam detection: a preliminary study," in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, pp. 25–28, Beijing, China, 2008.

[14] A. Benczúr, I. Bíró, K. . Csalogány, and T. Sarlós, "Web spam detection via commercial intent analysis," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, pp. 89–92, Banff, Canada, May 2007.

[15] I. Bíró, D. . Siklósi, J. . Szabó, and A. A. Benczúr, "Linked latent dirichlet allocation in web spam filtering," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pp. 37–40, Madrid, Spain, April 2009.

[16] Y. Liu, R. Cen, M. Zhang, S. Ma, and L. Ru, "Identifying web spam with user behavior analysis," in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, pp. 9–16, Beijing, China, April 2008.

[17] I. Luca, T. Kurt, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein, "Cloak of visibility: detecting when machines browse a different web," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 743–758, IEEE, San Jose, CA, USA, May 2016.

[18] G. Liu, X. Huang, X. Liu, and H. Fan, "Document sentiment modeling based on topic attention hierarchy memory

network," *Journal of Sichuan University. Natural Science Edition*, vol. 56, no. 5, pp. 55–64, 2019.

[19] S. H. Reza Mohammadi and M. A. Zare Chahooki, "Web spam detection using multiple kernels in twin support vector machine," 2016, https://arxiv.org/abs/1605.02917.

[20] J. Fdez-Glez, D. Ruano-Ordás, R. Laza, J. R. Méndez, P. Reyes, and F. Fdez-Riverola, "WSF2: a novel framework for filtering web spam," *Scientific Programming*, vol. 2016, Article ID 6091385, , 2016.

[21] Y. Mei, J. Zhang, J. Wang, J. Gao, T. Xu, and R. Yu, "The research of spam web page detection method based on web page differentiation and concrete cluster centers," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 820–826, Springer, Tianjin, China, June 2018.

[22] H. Jelodar, Y. Wang, C. Yuan, and X. Jiang, "A systematic framework to discover pattern for web spam classification," in *Proceedings of the 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 32–39, IEEE, Vancouver, Canada, November 2017.

[23] F. Asdaghi and A. Soleimani, "An effective feature selection method for web spam detection," *Knowledge-Based Systems*, vol. 166, pp. 198–206, 2019.

[24] M. S. Renato, T. A. Almeida, and A. Yamakami, "Artificial neural networks for content-based web spam detection," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, July 2012.

[25] Y. Li, X. Nie, and R. Huang, "Web spam classification method based on deep belief networks," *Expert Systems with Applications*, vol. 96, pp. 261–270, 2018.

[26] A. Makkar, M. S. Obaidat, and N. Kumar, "FS2RNN: feature selection scheme for web spam detection using recurrent neural networks," in *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Abu Dhabi, UAE, December 2018.

[27] A. Belahcen, M. Bianchini, and S. Franco, "Web spam detection using transductive (inductive graph neural networks," in *Advances in Neural Networks: Computational and Theoretical Issues*, pp. 83–91, Springer, Berlin, Germany, 2015.

[28] H. Fu, X. Xie, Y. Rui, N. Z. Gong, G. Sun, and E. Chen, "Robust spammer detection in microblogs," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 6, pp. 1–31, 2017.

[29] N.'A. Maulat Samsudin, C. F. Mohd Foozy, N. Alias, P. Shamala, N. F. Othman, and W. I. Sofiah Wan Din, "Youtube spam detection framework using naïve bayes and logistic regression," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1508–1517, 2019.

[30] E. Ezpeleta, M. Iturbe, I. Garitano, I. Velez de Mendizabal, U. Zurutuza, and António Sáez, Edited by A. Enrique, de la Cal, Á. Herrero, and H. Quintián, Eds., "A mood analysis on youtube comments and a method for improved social spam detection," in *Hybrid Artificial Intelligent Systems*, E. Corchado, Ed., pp. 514–525, Springer International Publishing, Cham, Switzerland, 2018.

[31] Laboratory of Web Algorithmics (http://law.di.unimi.it/)s University of Milan. Web collection uk-2006/uk-2007. https://github.com/keras-team/keras (2019).

[32] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Sunnyvale, CA, USA, November 2004.

[33] M. Kusner, Yu Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proceedings of the International Conference on Machine Learning*, pp. 957–966, Lille, France, July 2015.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[35] Beautiful soup documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc (2020).

[36] Googlenews-vectors-negative300. https://code.google.com/archive/p/word2vec/(2019).

[37] Document object model-introduction to the dom. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model (2020).

[38] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 161–168, New York, NY, USA, June 2006.

[39] K. L. Goh and A. K. Singh, "Comprehensive literature review on machine learning structures for web spam classification," *Procedia Computer Science*, vol. 70, pp. 434–441, 2015.

[40] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[42] Keras: Deep learning for humans. https://github.com/keras-team/keras (2019).

[43] M. Abadi, A. Agarwal, B. Paul et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," 2016, https://arxiv.org/abs/1603.04467.

[44] Web spam challenge phase iii results. http://webspam.lip6.fr/wiki/pmwiki.php?n=Main.PhaseIIIResults (2019).

[45] H. Wahsheh, I. Abu Doush, M. Al-Kabi, I. Alsmadi, and E. Al-Shawakfa, "Using machine learning algorithms to detect content-based Arabic web spam," *Journal of Information Assurance & Security*, vol. 7, no. 1, 2012.

[46] S. Mittal and A. Juneja, "Feature selection-model-based content analysis for combating web spam," *Computer Science & Information Technology*, vol. 6, pp. 27–34, 2016.

[47] S. Seabold and J. Perktold, "Statsmodels: econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, Austin, TX, USA, July 2010.

[48] X. Robin, N. Turck, H. Alexandre et al., "proc: an open-source package for r and s+ to analyze and compare roc curves," *BMC Bioinformatics*, vol. 12, no. 1, pp. 1–8, 2011.

[49] C. Chen, A. Liaw, L. Breiman et al., *Using random forest to learn imbalanced data*, p. 24, University of California, Berkeley, CA, USA, 2004.