Int. J. Nonlinear Anal. Appl. 11 (2020) No. 1, 321-339
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2020.4297

A Novel Set of Contextual Features for Web Spam Detection

Faeze Asdaghi*, Ali Soleimani and Morteza Zahedi

Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran;

(Communicated by Madjid Eshaghi Gordji)

Abstract

Web spam is one of the significant problems facing search engines. It wastes sources and time, decreases the quality of results and leads to user discontent. The two main approaches to the detection spam web pages are link and content-based analysis. In this study, we mainly focus on content-based analysis in both user-visible text and the source code of a web page to propose a set of features for web spam detection. we explore the relationship between types and frequency of HTML (HyperText Markup Language) tags used in a web page source code. We also examine the structure of the URL as the other source of information. Finally, the content of a web page visible to the user is considered semantically in order to identify relevance among the number of the existing topics in the text as well as the coherence of a text using Latent Dirichlet Allocation. Experimental results show that the proposed features increases the index of balanced accuracy from 0.33 to 0.69 and improves the web spam detection rate.

Keywords: web spam, content-based features, URL structure, HTML tags, topic modeling, Latent Dirichlet Allocation.
2010 MSC: 68T10,68T20

1. Introduction

Web spam refers to a host of techniques to subvert the ranking algorithms of web search engines, thereby raising the rank of search results. Examples of such techniques include content spam (filling web pages with popular and often highly monetizable search terms), link spam (creating links to a

Faeze Asdaghi
Email address: asdaghi@shahroodut.ac.ir, solimani_ali@shahroodut.ac.ir, zahedi@shahroodut.ac.ir (Faeze Asdaghi*, Ali Soleimani and Morteza Zahedi)

Received: October 2019 *Revised:* January 2020

page in order to increase its link-based score), and cloaking (serving different versions of a page to search engine crawlers rather than to human users). Web spam is annoying to search engine users and disruptive to search engines; therefore, most commercial search engines try to combat web spam [1].

The first step in combating web spam is its detection. It can be considered as ranking or machine learning problems. In the ranking problem, the confidence of a web page is measured based on the reputability of its neighbor in the web graph. In these methods, links and connections between web pages are more important than their content. Hyperlink-Induced Topic Search (HITS) and TrustRank are major algorithms in this category [2, 3, 4, 5]. This type of algorithm is sensitive to user queries and induces a web graph by finding a set of pages by searching a given query string. The main drawback is that it favors older pages, as a new page, even an excellent one, does not have many links unless it is part of an existing web site. Also, since it is query-dependent, the query time evaluation will be expensive.

However, in the machine learning approach, the content is essential. These methods seek for discriminative features and high-performance algorithms to detect spam pages. Common classification algorithms, such as decision trees, Support Vector Machines (SVM), etc., are widely used for this purpose [6] [7, 8]. Also, ensemble methods are employed to improve their performance [9] [10]. Research shows that machine learning approaches yield more accurate results than ranking approaches. However, classifiers need to be trained with new data when a new kind of spam is identified. Also, some methods like Artificial Neural Networks (ANN) demands significant computing time for spam classification. To achieve the goals of high accuracy and low time consumption requires sustained efforts to strike a balance between functionality and effectiveness [11].

In the field of feature, there are two main approaches, including link-based and content-based features. Link-based features investigate the properties of hyperlinks in web pages and pages linked to them. For example, the number of recovered links, the number of sites pointing to the analyzed site and the number of external and internal links are introduced as new features in [12]. The content-based feature is attentive in many aspects. In [13] and [14], some statistical features like the number of words on the page, title, and anchor text, length of words and rate of compression are discussed.

In this study, we focus on the role of features in improving the performance of classification and investigate some of the most common features including content-based, link-based, and direct features. In addition, we propose new features based on HTML tags and text coherence. The major contributions made by this paper are as follows:

- Introducing two groups of the novel, highly discriminative and computationally inexpensive feature subset to have learning classifiers deliver superior performance with regards to the detection of web spam. Our proposed features provide broader coverage and, consequently, higher accuracy compared to usual features.

- A classifier performance measure, which is suitable for unbalanced data, is presented and compared to other performance measures.

- The effectiveness of novel features in learning the classifier is shown by conducting preliminary experiments on standard WEBSPAMUK2007 [15] benchmark datasets.

The rest of this article is organized as follows. In Section 2, the related literature is reviewed. Section 3 describes the proposed features. Section 4 provides an experimental evaluation of the proposed novel web spam features on a standard dataset. Finally, Section 5 presents the concluding remarks and future research directions.

2. Preliminaries

In this section, we discuss the types of existing extracted features and methods of selecting the discriminating ones along with an explanation of the used features.

2.1. Feature Extraction

The first step of the machine learning approach is feature extraction. Extracted features in web spam detection are divided into three major groups: direct features, content-based features, and link-based ones.

Direct features. This type of feature is content-independent. It is extracted from primitive information about a web page that is gained without visiting a page or inspecting its content. Features like the number of pages in the host and the number of characters in the host name belong to this category. The number of these features is limited, and they are often merged in link-based features.

Link-based features. This type of feature is content-independent. It is extracted from primitive information about a web page that is gained without visiting a page or inspecting its content. Features like the number of pages in the host and the number of characters in the host name belong to this category. The number of these features is limited, and they are often merged in link-based features. These features can be classified into raw and transformed forms. It includes mostly the ratios of features such as Indegree/PageRank or TrustRank/PageRank and logarithm of several features. The transformation is more useful for classification than the raw link-based features.

Content-based features. This type of feature is computed from the content of a web page (whether raw text or user-viewed content). Some of these features include the number of words, average word length, and the average length of the title. Like link-based features, these features can also be computed for both the home page and other pages.

In addition to standard features, some different features have been introduced by different authors, some of which are listed in Table 2. These features focus on different parts of a web page, trying to find attributes of standard pages using diverse methods ranging from language models to the spelling check.

In this paper, we seek to introduce some new features based on the semantic relationship between the content of a web page. To achieve this purpose, we use topic modeling, which is a statistical model for discovering the abstract "topics" that appear in a collection of documents. Typically, algorithms such as Vector Space Model (VSM), Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA) are applied for topic modeling. In this study, we have used LDA.

Latent Dirichlet Allocation Latent Dirichlet Allocation (LDA) is a probabilistic model generated from a dataset. This method is based on the premise that documents are a mixture of random latent topics, which are characterized by a distribution over words. The words with the more probabilities in each topic usually give a general idea about the specific topic. It is a Bayesian inference model that uses Dirichlet distributions as prior distributions for the document-topic and word-topic distributions, lending itself to better generalization [15]. In a corpus D consisting of M documents, with document d having N_d words ($d \in 1, \ldots, M$), LDA models D according to the following generative process [15]:

- Choose a multinomial distribution ϕ_t for topic $t(t \in 1, \ldots, T)$ from a Dirichlet distribution with parameter β.

Table 1: List of innovative features for web spam detection.

Author(s)	Year	Features
Wang et al. [13]	2007	Amount of anchor text The fraction of visible content The fraction of globally trend words Outliers in the distribution of in-degrees and out-degrees of the graph induced by web pages and their hyperlinks The evolution rate of web pages in a given site Excessive replication of content
Jakub et al. [16]	2008	The fraction of unique POS to all available POS in the web page The sum of the POS entropy of a web page using 2-gram The fraction of the sum of unique verbs and nouns to total words The fraction of number of pronouns in a page to all words
Martinez et al. [17]	2009	The relationship between title and content, anchor text, and tokens of referring URL, meta tag description, and content/title of a web page using the Kullback-Liebler language model.
Wang et al. [12]	2010	Date of the last update The fraction of broken links to all links The fraction of Sum of token length to the text length
Pavlov et al. [18]	2011	Number of past tense verbs in the text The average number of punctuation signs The ratio of words with more than seven/ less than three characters to all words
Prieto et al. [19]	2012	Direction in meta tag Number of spelling and grammatical mistakes
Karunakaran et al. [20]	2014	Calculating text similarity using Jansen-Shannon language model
Luckner et al. [14]	2014	The ratio of meaningless tokens in the text Value of Gunning Fog Index
Hunagund et al. [21]	2015	Number of stop words in the title and text
Kumar et al. [22]	2016	The ratio of exciting words used in the text Existence of repeating patterns/ similar codes
Singh et al. [23]	2017	Density of keywords The fraction of the sum of Pertinence factors to total words in a web page

- Choose a multinomial distribution θ_d for document $d(d \in 1, \ldots, M)$ from a Dirichlet distribution with parameter α.
- For a word $w_n(n \in 1, \ldots, N_d)$ in document d, Select a topic z_n from θ_d. Select a word w_n from ϕ_{z_n}.

In the above generative process, only words in documents are observed variables, while other words are latent variables (ϕ and θ) and hyperparameters (α and β). In order to infer latent variables and hyperparameters, the probability of observed data D is computed and maximized as follows:

$$\prod_{d=1}^{M} \int p(\theta_d|\alpha)(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(Z_{dn}|\theta_d)p(w_{dn}|z_{dn}, \beta))d\theta_d \tag{2.1}$$

Defined α parameters of topic Dirichlet prior and the distribution of words over topics, which are drawn from the Dirichlet distribution, given β. T is the number of topics, M is the number of documents; N is the size of the vocabulary. The Dirichlet-multinomial pair for the corpus-level topic distributions, considered as (α, θ). Also, (β, ϕ) is the Dirichlet-multinomial pair for topic-word distributions. θ_d represents document-level variables sampled when per document. $z_d n$, $w_d n$ are word-level variables and sampled when for each word in each text document.

2.2. Feature Selection

Feature selection is the process of eliminating irrelevant and redundant variables in order to make sense of data, reduce computation requirement, decrease the effect of dimensionality curse, and improve the predictive performance. The feature selection is designed to choose an optimal subset of features from the input data, in an attempt to provide good prediction results or optimize the value of an evaluation function. There are several methods, such as Exhaustive, Best First, Genetic, Greedy, and Forward Selection Algorithm used in the web spam detection field. In [24] and [25], the correlation coefficient analysis is applied to the feature selection. A wrapper feature selection method in [8] selects a discriminating feature subset that evaluates features by a learning model. Authors in [14] have employed a classification tree that selects features based on the Gini coefficient to estimate the significance of features. Another feature selection method is the univariate filter feature selection. As in filter-based feature selection, features are selected independent of the induction algorithm, the measurement for feature selection is chosen as Mutual Information Maximization (MIM) [26]. Recently, the effect of feature selection on the rate of web spam detection has been evaluated [27]. In this study, more than 20 different methods were considered, and finally, a new algorithm (Smart-BT) was proposed. It was a backward elimination feature subset selection method based on measuring the effect of eliminating a set of features on the performance of a classifier rather than a single feature used in the sequential backward selection. The authors demonstrated its efficiency compared to other methods. The algorithm is shown in Figure 1. Considering the above, we used Smart-BT in the proposed model.

3. The Proposed Algorithm

In this section, we introduce some content-based features that are could be used to improve the detection rate of web spam. These features are based on HTML tags in a web page source code, Uniform Resource Locator (URL) structure and meaningfulness of the page content to propose a model for web spam detection. The inspiration and mathematical modeling of this model are described in detail.

3.1. Features based on URL structure and HTML tags

3.1.1. Inspiration

The main idea of this model is that each feature vector is an n-dimensional vector with numerical features that represents some objects and is treated as random variables and vectors, respectively, and their distributions depend on the nature of that object. A random variable $X : \Omega \to E$ is a measurable function on a set of possible outcomes Ω within a measurable space E. The probability that X assumes a value from a measurable set $S \subseteq E$ is calculated from formula (1) where P is the probability measure equipped with Ω.

$$Pr(X \in S) = P(\omega \in \Omega | X(\omega) \in S) \tag{3.1}$$

In classification, a good feature is the one with a high probability of occurrence in the distinct intervals of each class. The probability density function (PDF) is used for this reason as follows:

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{3.2}$$

By knowing such information about each feature, we can find an interval with the highest probability of occurrence. If this interval is distinct in data belonging to each class, it can be concluded

```
Algorithm finding unvalued features
input:  feature set of dataset
output: feature set Result such that removing it from dataset leads achieving BestIBA
Initialize the Irremovable and LowInfoFeatures set to Null, i to 1 and BestAnswer to 0
    BaseIBA = Evaluate (AllFeatures)
    /* calculate IBA of dataset using input features */
    CreateSubSet (WorkingSet, i)
    /* Create all i-member subset of features set and put it in WorkingSet */
    while ( WorkingSet(i) is not null) do
        for each ws ∈ WorkingSet do
            if (exist ir ∈ Ivvremovable that ir is a subset of ws) then
                Eliminate(ws)
                /* eliminates ws from WorkingSet */
            else
                e := Evaluate(AllFeatures - ws)
                diff := BaseIBA - e
                if (e > 0)
                    Remove (ws, Ivvremovable)
                    /* removes ws to Ivvremovable set */
                else
                    Remove (ws, LowInfoFeatures)
                    if (e >= BaseIBA) then
                        BestIBA := e
                        Result := ws
                    end if
                end if
            end if
        end for
        i := i + 1
        CreateSubSet (WorkingSet, i)
    end while
```

Figure 1: The pseudo-code of Smart-BT.

that the feature is suitable for classification, and the input data could be labeled with the correct class. Thus, we used this function to investigate new features and measure their value.

3.1.2. List of Features based on URL structure

As discussed in [28], the lexical features surrounding a URL are conductive to spam detection. The number of subdomains, length of a URL, and terms that appear in a URL allow a classifier to distinguish between "get.cheap.greatpills.com" and "google.com". A hostname is a combination of the host's local name and its parent domain's name. For example, the host "mail.google.com" consists of a local name "mail' and the domain name "google.com".

Number of subdomains. Spammers usually create multiple sub-domains on a single domain to create several websites. It is aimed to reduce the cost of purchasing multiple domains and hosting charges. Thus, spam pages are more likely to be hosted on a subdomain. Therefore, "." number is a good indicator of quality assessment. As shown in Figure 2, above 70% of all URLs contain more than

two and less than 5 ".". If a URL has more than five dots, it probably belongs to the spam category. The horizontal axis shows the number of dots in the URL.

Number of digits and special characters in domain and URL. A domain name that contains several digits or/and special characters is less likely to belong to a user-friendly domain name. It may have been created by automated software only to create link farms. As depicted in Figure 2, most of the regular pages have no digit or special character in the domain tokens. In this figure, the horizontal axis shows the number of digits and special characters in a domain.

Length of domain and URL. Spam pages tend to have long URLs due to keyword stuffing in the URL. An example of such a website is http://www.buycheapextralongshowercurtains.com. Thus, there is a high probability that the length of URLs represents spam. According to [29], short URLs are preferred by most search engines. As shown in Figure 2, it is clear that the URL length of most normal pages is less than 20. In this figure, the horizontal axis shows the URLs' length.

The average length of tokens in domain and URL. A URL could split into components such as the domain, path, and query parameters. Tokenization is conducted by splitting a URL into characters like dot and slash. On a regular page, tokens are almost valid words, and their length is within a specific range, but a spam page concatenates words to keyword stuffing. As depicted in Figure 2, the average length of tokens in both domain and path on a regular page is shorter than spam ones.

Consecutive character ratio in domain and URL. Spammers usually register domains in bulk using automated software. Web sites hosted on these domains are not suitable for humans, as they are designed for search engine crawlers to get a high PageRank. As a result, they are only available for a short time. These types of domains usually contain a combination of letters and numbers that create meaningless expressions. Therefore, a domain name containing a token with a combination of numeric and alphabetic letters is probably spam. To distinguish such domains, we propose an index called "consecutive character ratio". It measures the length of the largest alphabetic and numeric substring and divides the results by the total length of the token. A larger number is less likely to be automatically generated and thus identified as spam, as shown in Figure 2.

Top domain similarity ratio in domain tokens. One of the spammer tricks involves using popular domain names as a subdomain or a substring in URL tokens. For example, a regular user may believe that the URLs, like 'microsoft.com.phishy.net', is related to microsoft.com. Therefore, the similarity of URL tokens to famous domain tokens could be a suitable indicator to recognize spam pages. The criterion to consider a website as a well-known is its rank in Alexa, an organization that determines the ranking of websites based on their traffic. We used the list of the top 500 most visited websites as favorite domains expected that they would not appear in token of regular page URLs. As illustrated in Figure 2, the tokens of regular pages barely correspond to popular websites' names.

3.1.3. List of Features based on HTML Tags

The HTML code of a web page is a suitable choice for extracting features. Using less or more than usual elements in spam web pages may increase or decrease the utilization of some HTML tags. In this section, we explore features that seem to play a role in identifying spam web pages.

A/ Link. In HTML, <A> and <link> tags are used to create a link to an external source or document. <Link > tag is used to define a link between a document and an external resource, specifying a connection to another document used in the <head> section. However, the <a> tag is employed to define a hyperlink. The number of <a> tags in the HTML code of a web page indicates

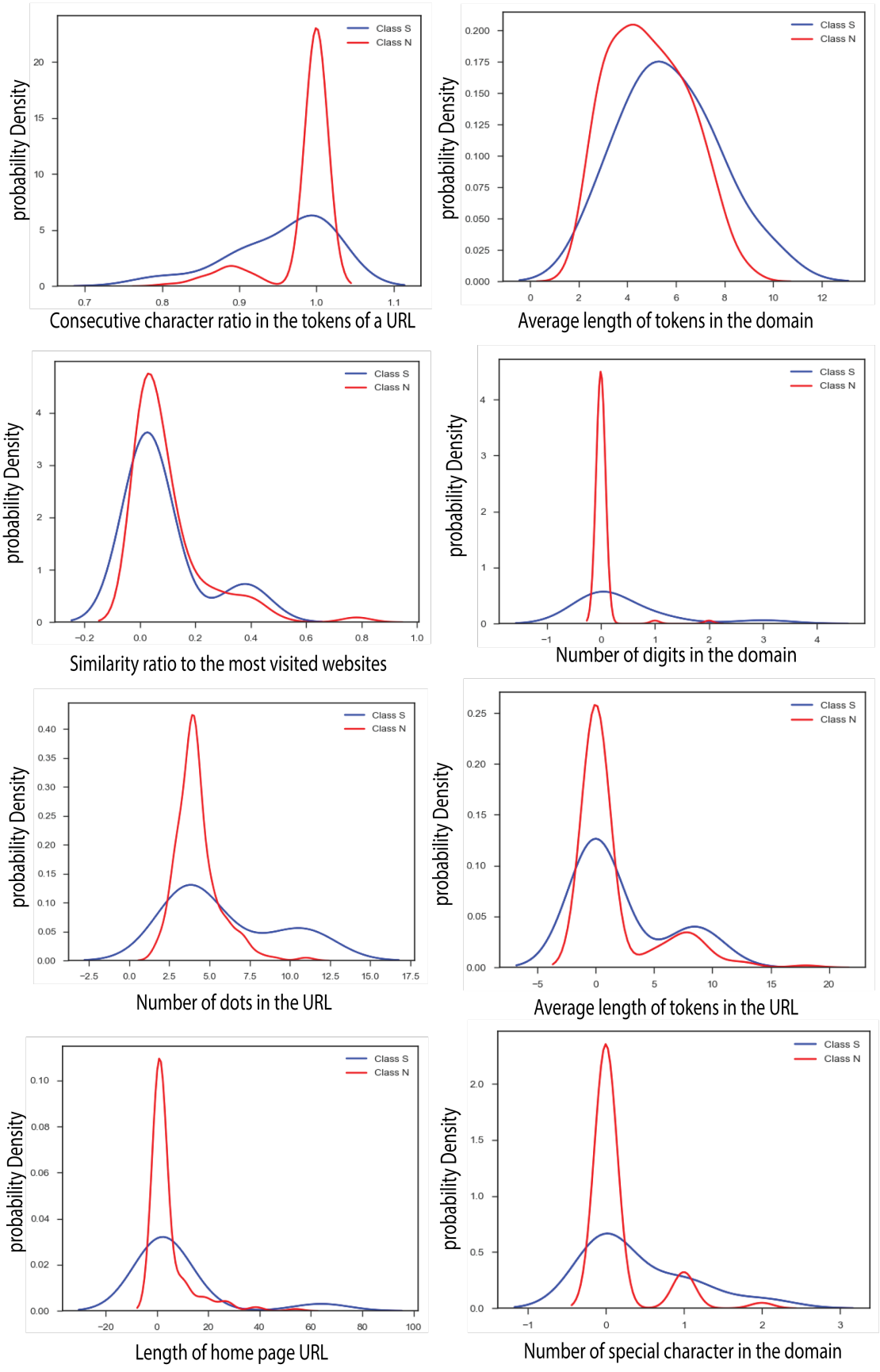

Figure 2: The probability distribution of URL based features.

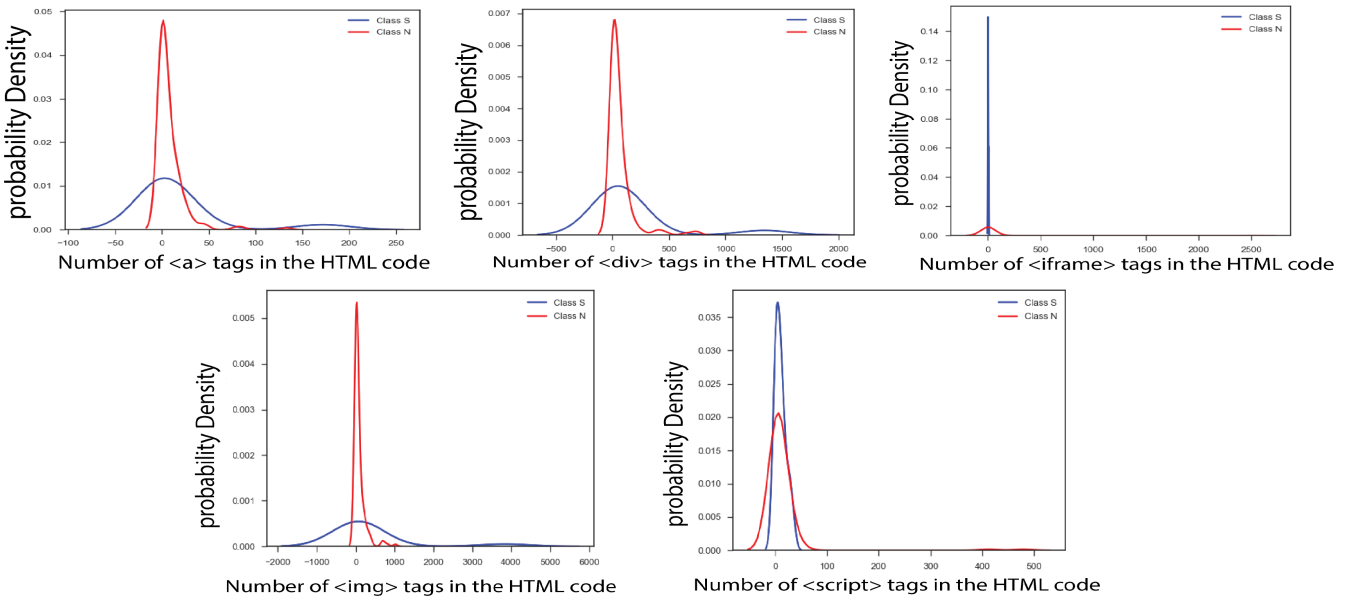

Figure 3: The probability distribution of HTML based features.

the number of links within the page regardless of their destination. Unlike <link>, this tag is not an empty element, specifying an object to be created on the page -like a clickable link or image, which redirects the user to another location. The <a> tag is only used in the <body> section. In spam pages, the number of links to other pages is greater than normal pages. As illustrated in Figure 3, we expect the number of these tags to be a good indicator of web spam detection.

Div. The tag is used to define a division or a section in an HTML document. It is like a container unit that encapsulates other page elements and divides the HTML document into different sections. Web developers utilize elements to group HTML elements, applying CSS styles to many elements at the same time. As Figure 3 shows, this tag is rarely used in common domains.

Iframe. The <iframe> tag is used to specify an inline frame to embed some other documents within the current HTML document. It defines a rectangular area within the document where the browser can display a separate document. This tag does not usually appear in spam pages (Figure 3).

Img. The <img> tag is used to define an image in an HTML page. Since images link to HTML pages, the <img> tag creates a holding space for the referenced image. Spam pages tend to have more images than regular pages. As Figure 3 shows, this tag is rarely seen in these pages.

Script. The < script > tag defines a client-side script (JavaScript) and contains scripting statements. Typical applications of JavaScript include image manipulation, form validation, and dynamic changes of content. Spam pages use Java Scripts codes to redirect visitors to other pages or to display advertisements and other attractive boxes.

3.2. Features based on Text Readability

3.2.1. Inspiration

One characteristic of spam web pages is low text quality. It is due to random text generation or collection of texts from different sources without considering the integrity of the final content.

Therefore, this can serve as a criterion to distinguish between spam and standard pages. For this purpose, we examine text coherence and cohesion. A coherent text can be described as a text where the information is organized and linked to a logically-connected unit with cohesive devices joining the parts so that the text makes sense. These cohesive devices, including conjunctions, reference words, substitution, as well as lexical devices such as repetition of words, collocations, and lexical groups, contain phrases or words that help the reader associate previous statements to subsequent ones. The higher application of these devices in a text enhances its coherence and readability. In [30], the authors identify five general categories of cohesive devices that signal coherence in texts:

- Reference: including Pronominal, Demonstrative, and Comparative References.
- Ellipsis
- Substitution
- Lexical: including Reiteration, Synonymy, and Hyponymy
- Conjunction: including Additive, Adversative, Causal, and Temporal Conjunctions

Furthermore, a text may be cohesive without necessarily being coherent. That is, cohesion does not guarantee coherence. Cohesion is determined by lexical and grammatical relationships between sentences, whereas coherence is reflected in semantic relationships. In other words, coherence is a semantic property of discourse established through the interpretation of each sentence relative to other sentences, with the word "interpretation," implying an interaction between the text and the reader. One way of evaluating coherence in a text is the topical structure analysis [31].

We assume that coherence is a condition that limits the number of topics in a text. It suggests that the topics of consecutive paragraphs are related and belong to the same field. For this reason, the number of topics in a text is a feature that could be used for detecting spam pages. Latent Dirichlet Allocation (LDA) is a popular algorithm for determining the number of text topics. As explained in Section 2, it finds latent topics in the corpus, indicating the probability of each topic appearing in the input text.

3.3. List of Features based on Coherence

In this section, we examine the effect of some cohesive devices with the low-cost calculation for detecting spam web pages. Reviewed items include the number of words, sentences, and paragraphs, as well as the number of conjunctions used in the text. The results are shown in Table 2.

3.4. List of Features based on Cohesion

In this part, we investigate text cohesion using topic modeling methods such as VSM, LSI, PLSA, and LDA and extract some features based on these methods. Vector Space Model is a simple model based on linear algebra that allows determining the degree of correspondence between queries and documents. However, it is high dimensional as it involves using vector space that is typically sparse, and therefore, the cosine similarity can be noisy and inaccurate. It also has to deal with the issue of polysemy and synonymy. Latent Semantic Indexing can handle the problem of synonymy to some extent, mapping documents onto a low dimensional space. It involves decomposing the term-document matrix to make it faster than other dimensionality reduction models. However, it is not as efficient in tackling the polysemy problem and is computationally intensive due to the application of Singular Value Decomposition (SVD). Also, it is difficult to update when new documents appear. Probabilistic Latent Semantic Analysis can overcome the polysemy problem, treating topics as word

Table 2: Features based on text coherence.

Description	Example
The ratio of basic connectives to all words	for, and, nor
The ratio of conjunctions to all words	and, but
The ratio of disjunctions to all words	Or
The ratio of simple subordinators to all words	after, although, as
The ratio of coordinating conjuncts to all words	yet, so, nor
The ratio of addition words to all words	and, also, besides
The ratio of sentence linking words to all words	nonetheless, therefore, although
The ratio of order words to all words	to begin with, next, first
The ratio of reason and purpose words to all words	therefore, that is why, for this reason
The ratio of opposition words to all words	but, however, nevertheless
The ratio of determiners to all words	a, an, the
The ratio of positive causal connectives to all words	arise, because, enabling
The ratio of positive logical connectives to all words	actually, after all, all in all
The ratio of temporal connectives to all words	a consequence of, after, again
The ratio of positive, intentional connectives to all words	by, desire, desired
The ratio of positive connectives to all words	actually, after, again
The ratio of demonstratives to all words	this, that, these
The ratio of additive connectives to all words	after all, again, all in all
The ratio of causal connectives to all words	although, arise, arises

distribution by using probabilistic methods instead of matrices. However, the number of parameters increases linearly relative to the number of documents. Latent Dirichlet Allocation utilizes Dirichlet priors for document-topic and topic-word distributions. It prevents over-fitting. It has a low computational cost and can be updated when new documents appear. Therefore, as discussed earlier, we use LDA to extract the number of topics raises in documents as a feature [32].

Before the application of LDA, we need to determine the number of topics in the dataset. To determine the optimal number of topics to be extracted by the LDA, the topic coherence score is usually recruited to assess the suitability of the extracted topics, where w_i, w_j denote top words in the topic:

$$CoherenceScore = \sum_{i<j} score(w_i, w_j) \tag{3.3}$$

$$Score(w_i, w_j) = log \frac{(p(w_i, w_j))}{(p(w_i)p(w_j))} \tag{3.4}$$

Probabilities are estimated based on word co-occurrence counts. These counts are derived from documents constructed with a sliding window that moves over Wikipedia and considered as an external reference corpus. The position of each window defines such a document. The coherence score estimated for the LDA model in our dataset is depicted in Figure 4. As displayed in the chart, the optimum number of topics in this data set is 38. Based on this model, we investigate some features for each document (web page):

- Number of topics with probability greater than 1%
- Number of topics with probability greater than 10%

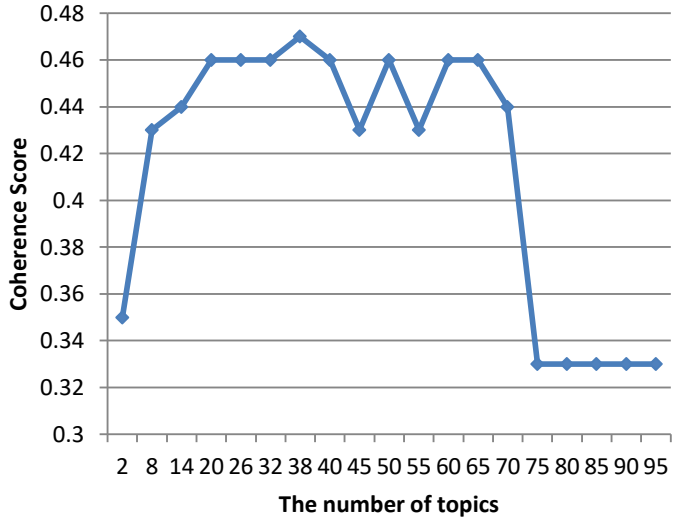

Figure 4: Impact of the number of topics on the model coherence score.

- Number of topics with probability greater than 20%
- Number of topics with probability greater than 30%
- Number of topics with probability greater than 40%

3.5. Proposed Model

As mentioned in the previous chapter, preprocessing the initial set of features consists of two main components of new features extracting, followed by the evaluation and selection of the chosen ones. In the first step, we describe a group of low-cost features based on the URL structure and HTML tags, investigating their value using the probability density function. Then, applying an appropriate feature selection method based on backward elimination, as introduced in [27], we decrease the number of features. The diagram of this algorithm is shown in Figure 5.

In this model, the input is the web page, which consists of four components: relation graph, source code, URL, and visible text. Each component is a good source of information and will be processed to extract the proper features. After extracting features, they are ranked according to their score using chi-square. In the next step, a subset of n best features is sent to the Smart-BT algorithm as the input data. The output is the final feature set utilized for classifying new web pages. We used Naïve Bayes as the classification algorithm of this model due to its low computational cost and excellent performance in binary classes, especially when one class is smaller than the other.

The heart of this model is the feature extraction part. As discussed in the literature review, there is a body of research that uses the link and content-based features. However, the emphasis of this paper is on extracting features based on HTML tags and URLs. In the next section, we introduce some features that are presumed to increase the detection rate of web spam. The probability density function is employed to identify distinct intervals of features with the highest probability of occurrence in each class in order to asses their usefulness. It is a kind of Histogram that applies kernel smoothing to plot values, with the peaks of the chart indicating the concentration of values over the interval. One advantage of this method over histograms is its more exceptional ability in determining the distribution shape as it is not affected by the number of bins (each bar used in a typical histogram), which are essential to keeping or discarding a feature.

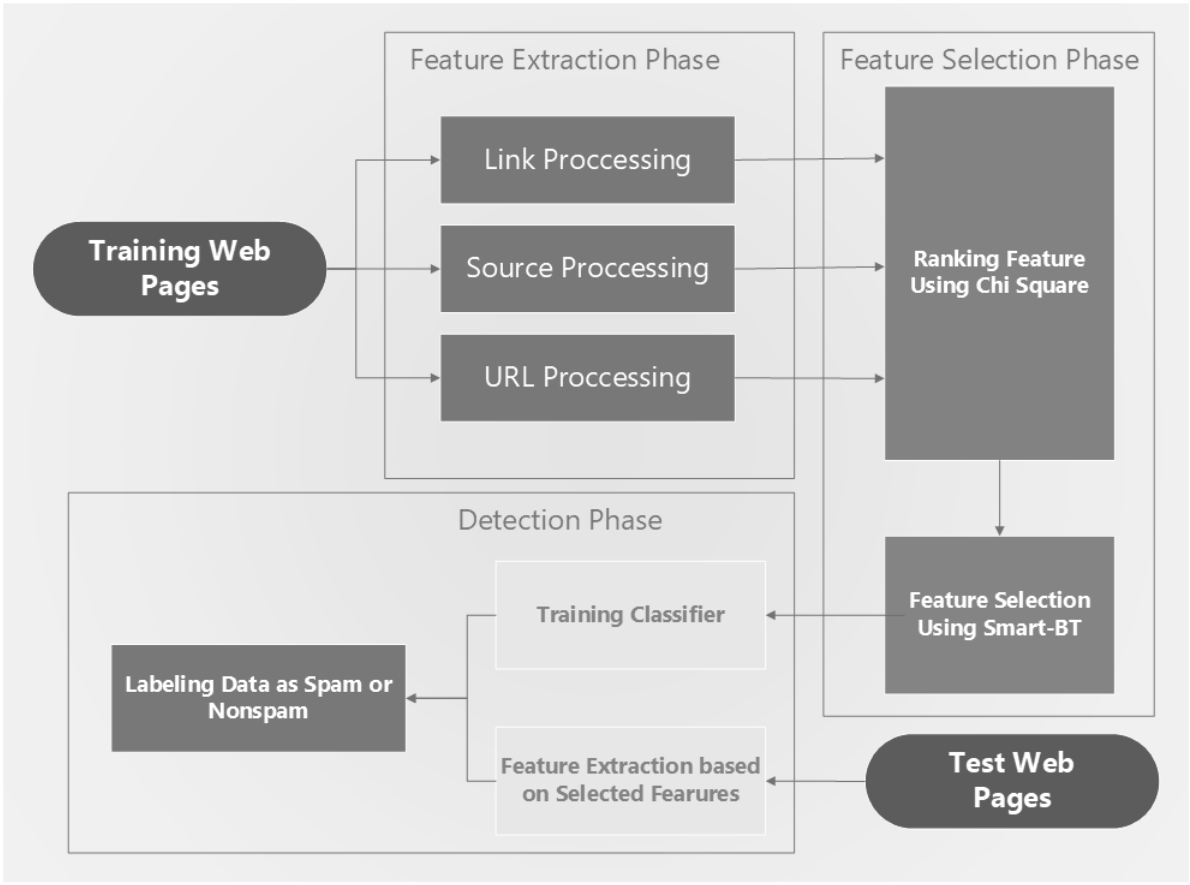

Figure 5: Diagram of the proposed model.

3.6. Computational Complexity

The proposed model consists of three phases:

- Feature extraction: since the time complexity may vary depending on the source of extracting features, we choose the worst ones.

  URL based: Since there are limited characters in the URL of a web page ([33]), the time and memory complexity of the following method of features calculation is $O(1)$.

  HTML based: If the source of each web page consists of n characters, then the time and memory complexity of extracting the following features are in the order of $O(n)$.

  Text Readability based: if the corpus contains N pages and each page has n characters since the LDA model is initially created and there is no need for recreation, the time and memory complexity of extracting the following features is in the order of $O(Nn)$.

- Feature selection: It is only used once in the training phase, and its complexity has no impact on time and memory consumption.

- Detection: it uses naïve Bayes as a classification method with a complexity of $O(Nd)$ where d is the number of features.

Therefore, the final time and memory complexity is $O(Nd + Nn)$ that can be summarized as, $O(n)$.

Table 3: Statistics of WEBSPAM-UK2007 collection.

	# Spam Hosts	# Non-spam Hosts
Training Dataset	222	3766
Testing Dataset	122	1933

Table 4: Measures of binary classification.

	Formula	Evaluation of Focus
Precision	$\frac{TP+TN}{TP+TN+FP+FN}$	The overall effectiveness of a classifier
Recall	$\frac{TP}{TP+FP}$	Class agreement of data labels with positive labels given by the classifier
F-score	$\frac{2TP}{2TP+FP+FN}$	Relationship between positive labels of data and those given by a classifier
Specificity	$\frac{TN}{TN+FP}$	Effectiveness of classifier in identifying negative labels
AUC	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$	Classifier's ability to avoid false classification

4. Evaluation of the proposed approach

4.1. Dataset and Evaluation Measures

4.1.1. Dataset

In this study, we used the WEBSPAM-UK2007 data set, which contains 105.9 million pages and over 3.7 billion links with about 114,529 hosts. It is designed to crawl the ".uk" web domain and provides each page's content and links as well as human-assessed categories of each host (spam, non-spam, and borderline) [34]. Unbalancing is a critical feature of this data set. Just 5% of the pages in this data set are spam, and others are normal. Table 3 shows it clearly.

4.1.2. Evaluation Measures

Standard evaluation metrics in the binary classification field are accuracy, precision, recall, and F-score, specificity, and area under the curve (Table 4). Where TN is the number of negative samples categorized by the algorithm, TP is the number of positive samples categorized correctly by the algorithm and, FP, and FN are the number of positive and negative samples that are not correctly categorized by the algorithm, respectively. These metrics are suitable for measuring the performance of classification methods when the dataset is balanced. However, when there is an unbalanced dataset, another metric is required to valorize the class with fewer instances.

In [33], Garcia et al. introduced a novel metric to evaluate the performance of a classifier on an unbalanced data set, which is called the Index of Balanced Accuracy (IBA). Moreover, Balanced Accuracy Graph, as the area of a rectangular region in a two-dimensional space, is defined by the product of accuracies in each class that is called Gmean2 or Dominance. Dominance is a suitable measure to calculate the overall accuracy in imbalanced domains. It is due to its ability to assess how prevalent is the dominant class rate with others as follows:

$$IBA_{\alpha} = (1 + \alpha.Dominance)M \tag{4.1}$$

Table 5: Comparing the results of data classification using different feature sets in WEBSPAM-UK2007.

Feature Set	**# Features**	**Accuracy**	**Specificity**	**IBA**
Content-Based	96	14%	**0.90**	0.002
Link-Based	41	**92%**	0.05	0.029
Transformed Link-Based	138	82%	0.34	0.192
Content+Link+Transformed	275	70%	0.64	0.337
URL Structure Based	49	80%	0.33	0.173
HTML tags Based	72	36%	042	0.222
Text Readability	52	82%	0.21	0.311
All Features	448	76%	0.65	**0.694**

Table 6: Comparison of the results.

Method	**# Features**	**Classifier**	**Accuracy**	**Recall**	**F1**	**AUC**	**IBA**
Our method	48	Naïve Bayes	95.8%	0.751	0.786	0.811	0.694
Paper [35]	275	Minimum Description Length (MDL) classifier	94.7%	-	0.225	-	0.400
Paper [36]	275	C5.0 + SVM	-	0.442	0.41	0.673	-
Paper [37]	96	Bayes network	93%	-	0.341	0.844	-
Paper [38]	137	Deep Believe Network	-	0.81	0.81	0.91	0.710

$$Dominance = TP - TN \tag{4.2}$$

$$M = TP \times (1 - FP) \tag{4.3}$$

In [39], they also showed that Dominance has a strong effect on IBA, and alpha plays an essential role in justifying the final results. They suggested that small alpha values (i.e., 0.05) are required to diminish this effect. In [27], it has been shown that this index is suitable for evaluating the classifier performance in an unbalanced data set and web spam detection.

4.2. Experimental results and Analysis

The feature extraction phase is implemented using Python V.3. Feature selection and data classification are conducted using the Weka library [40]. The comparison was performed by WEBSPAM-UK2007 [34], discussed in the previous section. The Naïve Bayes method is utilized for classification experiments, and data is tested by a ten-fold cross over. For each data set, 14 URL structure-based and 24 HTML tags based features are calculated in three modes: (1) minimum values between all pages of a specific domain, (2) maximum values between all pages of a specific domain, (3) average values between all pages of a specific domain (total of 42 URL and 72 HTML tag based features). We also extract seven features of the domain name, including the number of dots, special characters, digits, length, similarity ratio, and the presence of "www" phrase or IP.

For content and link-based features, we also applied precomputed features introduced in the WEBSPAM-UK2007 dataset, which consists of 96 content-based, 41 link-based, and 138 transformed

link-based features. The use of these features without the removal of redundant and useless ones reduces the detection rate, as is shown in Table 5.

Table 7: List of selected features.

Type	**Features description**
Content	Top 100 and 200 queries recall (hp)
	Top 100 queries recall (mp)
	Top 500 corpus precision (mp)
	The fraction of visible text (average value of all pages in the host)
	Top 200, 500 and 1000 queries precision (average value of all pages in the host)
	Top 100 and 200 queries recall (average value of all pages in the host)
	Number of words in the page (standard deviation of all pages in the host)
	Average word length (standard deviation of all pages in the host)
	The fraction of anchor text (standard deviation of all pages in the host)
Link	Independent LH (standard deviation of all pages in the host)
	Truncated PageRank using truncation distance 1, 2, 3 and 4, mp
	Out-degree of hp and mp
	PageRank of mp
Transformed Link	The logarithm of truncated PageRank
	based on truncation distance 1, 2, 3 and 4, mp
	The logarithm of out-degree of hp and mp
	The logarithm of PageRank of mp
URL	Maximum number of dots in the URL path and domain
	Number of special characters in the domain and URL
	Minimum and average number of digits in the URL path
	Minimum and the average length of the URL path
	Minimum and the average length of path and domain tokens
	Average consecutive character ratio of URL tokens
	The highest similarity of domain tokens to popular domains
HTML Tags	Minimum number of <a> tags
	Maximum number of tags
	Average number of <img> tags
	Minimum number of <script> tags
	Number of <iframe> tags
	Number of <link> tags
Text Readability	Number of demonstratives
	Number of topics with probability greater than 1%
	Number of topics with probability greater than 10%
	Number of topics with probability greater than 40%

In this table, the accuracy, precision, recall, and IBA of data classification are calculated by the Naïve Bayes algorithm for each feature group. The best results are shown in bold. As can be seen, using all kinds of features yields higher performance. However, it imposes a high computational cost on the system due to a large number of features. Therefore, in the next step of the proposed model,

we rank features by the chi-square test and send chosen features to the Smart-BT algorithm as the input. The best result comprises 48 features, which increases IBA from 0.360 to 0.694. A comparison of the results with those reported by Silva et al. [35], Patils [6], Lu [10], and Fdez-Glez [36] is drawn in Table 6. As can be observed, the value of IBA in all three cases is relatively identical, but the number of features used to achieve these values is radically different. Also, the time complexity in our method is $O(n)$, while it is more in other methods, especially in the Deep Believe Network method. In addition, the chosen features are indexed in Table 7. In this table, "hp" denotes the home page, and "mp" indicates the page with the highest page rank.

5. Conclusion

In this study, a set of effective features is proposed for web spam detection. For this reason, three groups of highly discriminative and computationally inexpensive features based on HTML tags, URL structure, and text readability are introduced. Then, we use the Smart-BT feature selection algorithm for dimension reduction, and finally, we test chosen features using Naïve Bayes classifier on the WEBSPAMUK2007 dataset. The results of experiments demonstrate the excellent performance of these features, with IBA increasing from 0.337 to 0.694. Although some of the URLs and HTML based features have been used in earlier works, here we conducted a comprehensive study of all possible features that could be extracted from these two sources (whether previously introduced or new ones). Then we determined which ones are discriminative by using a recent efficient algorithm (Smart-BT). Also, due to the small number of chosen features and their low computational cost, it can be used in real-time applications or those with short running time. As the next step, it would be a good improvement using another low-cost classifier with the ability of enhancement occurring new data instead of Naïve Bayes.

References

[1] M. Najork, Web Spam Detection, in: Encyclopedia of Database Systems, Springer New York, 2017, pp. 1–5. doi:10.1007/978-1-4899-7993-3_465-3.
URL https://doi.org/10.1007{%}2F978-1-4899-7993-3{_}465-3

[2] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, R. Baeza-yates, L. Sapienza, Using Rank Propagation and Probabilistic Counting for Link-Based Spam Detection, Proc. of WebKDD 6 (2006). doi:10.1.1.60.6845.

[3] P. Boldi, M. Santini, S. Vigna, {PageRank} as a function of the damping factor, in: Proceedings of the 14th international conference on World Wide Web - {WWW} {\textquotesingle}05, {ACM} Press, 2005. doi:10.1145/1060745.1060827.
URL https://doi.org/10.1145{%}2F1060745.1060827

[4] H. Garcia-molina, J. Pedersen, Francis Clarke : biography (1973) (2004) 2004–2004.

[5] V. Krishnan, R. Raj, Web spam detection with antitrust rank, Proceedings of the 2nd Int. Workshop on Adversarial Information Retrieval on the Web, AIRWeb 2006 - 29th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR 2006 (2006) 37–40.

[6] R. C. Patil, D. R. Patil, Web spam detection using {SVM} classifier, in: 2015 {IEEE} 9th International Conference on Intelligent Systems and Control ({ISCO}), IEEE, 2015. doi:10.1109/isco.2015.7282294.
URL https://doi.org/10.1109{%}2Fisco.2015.7282294

[7] K. L. Goh, A. K. Singh, Comprehensive Literature Review on Machine Learning Structures for Web Spam Classification, Procedia Computer Science 70 (2015) 434–441. doi:10.1016/j.procs.2015.10.069.
URL https://doi.org/10.1016{%}2Fj.procs.2015.10.069

[8] A. A. Torabi, K. Taghipour, S. Khadivi, Web Spam Detection: New Approach with Hidden Markov Models, in: Information Retrieval Technology, Springer Berlin Heidelberg, 2013, pp. 239–250. doi:10.1007/978-3-642-45068-6_21.
URL https://doi.org/10.1007{%}2F978-3-642-45068-6{_}21

[9] L. Araujo, J. Martinez-Romo, Web Spam Detection: New Classification Features Based on Qualified Link Analysis and Language Models, {IEEE} Transactions on Information Forensics and Security 5 (3) (2010) 581–590. doi:

10.1109/tifs.2010.2050767.
URL https://doi.org/10.1109{%}2Ftifs.2010.2050767
[10] X.-Y. Lu, M.-S. Chen, J.-L. Wu, P.-C. Chang, M.-H. Chen, A novel ensemble decision tree based on under-sampling and clonal selection for web spam detection, Pattern Analysis and Applications 21 (3) (2017) 741–754. doi:10.1007/s10044-017-0602-2.
URL https://doi.org/10.1007{%}2Fs10044-017-0602-2
[11] K. Hans, L. Ahuja, S. K. Muttoo, Approaches for Web Spam Detection, International Journal of Computer Applications 101 (1) (2014) 38–44. doi:10.5120/17655-8467.
URL https://doi.org/10.5120{%}2F17655-8467
[12] W. Wang, G. Zeng, D. Tang, Using evidence based content trust model for spam detection, Expert Systems with Applications 37 (8) (2010) 5599–5606. doi:10.1016/j.eswa.2010.02.053.
URL https://doi.org/10.1016{%}2Fj.eswa.2010.02.053
[13] W. Wang, G. Zeng, Content Trust Model for Detecting Web Spam, in: {IFIP} International Federation for Information Processing, Springer {US}, pp. 139–152. doi:10.1007/978-0-387-73655-6_10.
URL https://doi.org/10.1007{%}2F978-0-387-73655-6{_}10
[14] M. Luckner, MichałGad, PawełSobkowiak, Stable web spam detection using features based on lexical items, Computers & Security 46 (2014) 79–93. doi:10.1016/j.cose.2014.07.006.
URL https://doi.org/10.1016{%}2Fj.cose.2014.07.006
[15] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, L. Zhao, Latent Dirichlet allocation ({LDA}) and topic modeling: models, applications, a survey, Multimedia Tools and Applications 78 (11) (2018) 15169–15211. doi:10.1007/s11042-018-6894-4.
URL https://doi.org/10.1007{%}2Fs11042-018-6894-4
[16] J. Piskorski, M. Sydow, D. Weiss, Exploring linguistic features for web spam detection: A preliminary study, AIRWeb 2008 - Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (2008) 25–28doi:10.1145/1451983.1451990.
[17] J. Martinez-Romo, L. Araujo, Web spam identification through language model analysis, in: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web - {AIRWeb} {\textquotesingle}09, {ACM} Press, 2009. doi:10.1145/1531914.1531920.
URL https://doi.org/10.1145{%}2F1531914.1531920
[18] A. Pavlov, B. Dobrov, Detecting content spam on the web through text diversity analysis, CEUR Workshop Proceedings 735 (2011) 11–18.
[19] V. M. Prieto, M. Álvarez, R. López-Garc\'\ia, F. Cacheda, Analysis and Detection of Web Spam by Means of Web Content, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 43–57. doi:10.1007/978-3-642-31274-8_4.
URL https://doi.org/10.1007{%}2F978-3-642-31274-8{_}4
[20] K. P. Karunakaran, S. Kolkur, Language Model Issues in Web Spam Detection, Enhanc. Res. Manag. Comput. Appl. 3 (2) (2014) 39–43.
[21] K. Hunagund, S. K. K. L, Spam Web Page Detection based on Content and Link Structure of the Site 4 (8) (2015) 8–11. doi:10.17148/IJARCCE.2015.4875.
[22] S. Kumar, X. Gao, I. Welch, Novel Features for Web Spam Detection, in: 2016 {IEEE} 28th International Conference on Tools with Artificial Intelligence ({ICTAI}), IEEE, 2016. doi:10.1109/ictai.2016.0096.
URL https://doi.org/10.1109{%}2Fictai.2016.0096
[23] T. Singh, M. Kumari, S. Mahajan, Feature oriented fuzzy logic based web spam detection, Journal of Information and Optimization Sciences 38 (6) (2017) 999–1015. doi:10.1080/02522667.2017.1372146.
URL https://doi.org/10.1080{%}2F02522667.2017.1372146
[24] A. H. Keyhanipour, B. Moshiri, Designing a web spam classifier based on feature fusion in the layered multi-population genetic programming framework, in: Proceedings of the 16th International Conference on Information Fusion, IEEE, 2013, pp. 53–60.
[25] M. Mahmoudi, A. Yari, S. Khadivi, Web spam detection based on discriminative content and link features, in: 2010 5th International Symposium on Telecommunications, IEEE, 2010. doi:10.1109/istel.2010.5734084.
URL https://doi.org/10.1109{%}2Fistel.2010.5734084
[26] S. Mittal, A. Juneja, Feature Selection Model Based Content Analysis for Combating Web Spam, in: Computer Science & Information Technology ({CS} & {IT}), Academy & Industry Research Collaboration Center ({AIRCC}), 2016. doi:10.5121/csit.2016.60403.
URL https://doi.org/10.5121{%}2Fcsit.2016.60403
[27] F. Asdaghi, A. Soleimani, An effective feature selection method for web spam detection, Knowledge-Based Systems

166 (2019) 198–206. doi:10.1016/j.knosys.2018.12.026.
URL https://doi.org/10.1016{%}2Fj.knosys.2018.12.026

[28] J. Ma, L. K. Saul, S. Savage, G. M. Voelker, Identifying suspicious URLs, in: Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09, ACM Press, New York, New York, USA, 2009, pp. 1–8. doi:10.1145/1553374.1553462.
URL https://doi.org/10.1145{%}2F1553374.1553462http://portal.acm.org/citation.cfm?doid=1553374.1553462

[29] K. Thomas, C. Grier, J. Ma, V. Paxson, D. Song, Design and Evaluation of a Real-Time URL Spam Filtering Service, in: 2011 IEEE Symposium on Security and Privacy, IEEE, 2011, pp. 447–462. doi:10.1109/SP.2011.25.
URL https://doi.org/10.1109{%}2Fsp.2011.25http://ieeexplore.ieee.org/document/5958045/

[30] M. Halliday, Cohesion in English, Routledge, 2014. doi:10.4324/9781315836010.
URL https://doi.org/10.4324{%}2F9781315836010https://www.taylorfrancis.com/books/9781315836010

[31] S. Hoenisch, Coherence and Cohesion in Text Linguistics.
URL https://www.criticism.com/da/coherence.php

[32] B. V. Barde, A. M. Bainwad, An overview of topic modeling methods and tools, in: 2017 International Conference on Intelligent Computing and Control Systems ({ICICCS}), IEEE, 2017. doi:10.1109/iccons.2017.8250563.
URL https://doi.org/10.1109{%}2Ficcons.2017.8250563

[33] V. Garc\'\ia, R. A. Mollineda, J. S. Sánchez, Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions, in: Pattern Recognition and Image Analysis, Springer Berlin Heidelberg, 2009, pp. 441–448. doi:10.1007/978-3-642-02172-5_57.
URL https://doi.org/10.1007{%}2F978-3-642-02172-5{_}57

[34] C. b. t. L. o. W. Algorithmics, Datasets.
URL https://chato.cl/webspam/datasets/

[35] R. M. Silva, T. A. Almeida, A. Yamakami, Towards Web Spam Filtering Using a Classifier Based on the Minimum Description Length Principle, in: 2016 15th {IEEE} International Conference on Machine Learning and Applications ({ICMLA}), IEEE, 2016. doi:10.1109/icmla.2016.0083.
URL https://doi.org/10.1109{%}2Ficmla.2016.0083

[36] J. Fdez-Glez, D. Ruano-Ordás, R. Laza, J. R. Méndez, R. Pavón, F. Fdez-Riverola, WSF2: A Novel Framework for Filtering Web Spam, Scientific Programming 2016 (2016) 6091385. doi:10.1155/2016/6091385.
URL https://doi.org/10.1155/2016/6091385

[37] M. D. Oskouei, S. N. Razavi, An Ensemble Feature Selection Method to Detect Web Spam, Asia-Pacific Journal of Information Technology and Multimedia 7 (2) (2018) 99–113.

[38] Y. Li, X. Nie, R. Huang, Web spam classification method based on deep belief networks, Expert Systems with Applications 96 (2018) 261–270. doi:10.1016/j.eswa.2017.12.016.
URL https://doi.org/10.1016{%}2Fj.eswa.2017.12.016

[39] V. Garcia, R. A. Mollineda, J. S. Sanchez, Theoretical Analysis of a Performance Measure for Imbalanced Data, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 617–620. doi:10.1109/ICPR.2010.156.
URL https://doi.org/10.1109{%}2Ficpr.2010.156http://ieeexplore.ieee.org/document/5597459/

[40] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The {WEKA} data mining software, {ACM} {SIGKDD} Explorations Newsletter 11 (1) (2009) 10. doi:10.1145/1656274.1656278.
URL https://doi.org/10.1145{%}2F1656274.1656278