Asian Journal of Information Technology 16 (7): 632-638, 2017
ISSN: 1682-3915

Web Spam Detection and Classification using Hybrid Extensive Machine Learning Algorithm (HEMLA) for Domain Specific Features

[1]T. Muralidharan, [2]V. Saishanmuga Raja and [3]S.P. Rajagopalan
[1]Department of CSE, Dr. MGR Educational and Research Institute University, Maduravoyal, Chennai, India
[2]Department of CSE, Shanmuganathan Engineering College, Arasampatti, Pudukottai, India
[3]Department of Computer Applications, Dr. MGR Educational and Research Institute University, Maduravoyal, Chennai, India

Abstract: In the most recent couple of years as internet utilization turns into the principle supply route of the life's every day exercises, the issue of spam turns out to be intense for web group. Spam pages frame a genuine risk for a wide range of clients. This risk demonstrated to advance constantly with no piece of information to lessen. Diverse types of spam saw an emotional increment in both size and negative effect. A lot of e-mails and website pages are considered spam either in Simple Mail Transfer Protocol (SMTP) or web crawlers. Numerous specialized strategies were proposed to approach the issue of spam. We propose a Hybrid Extensive Machine Learning Algorithm (HEMLA) for detection and classification of that offers weight to the data nourished by clients and thinks about the presence of some space particular components. Hybrid extensive machine learning algorithm is a combination of many learning algorithms like conjugate gradient, resilient back-propagation and levenberg-marquardt algorithms. The outcomes demonstrate that the hybrid extensive machine learning algorithm overcomes the traditional web filtering methods as far as reducing the false positives and the false negatives and increasing the accuracy.

Key words: Simple mail transfer protocol, back-propagation, information, spam, filtering methods, supply route

INTRODUCTION

With the expanded ways in web applications and the expansion of data accessible for people in general, the requirement for proficient web indexes that can recover the most significant archives that fulfill client's needs gets to be obvious. From Information Retrieval (IR) point of view, search engines are in charge of recovering an arrangement of reports that are positioned in dropping request as indicated by their importance (Yates and Neto, 1999). A typical issue experienced in this connection is that there are a few records set apart with a high rank and recovered as the first (or one of the top) reports by the search engines where they are genuinely not (Gyongyi and Molina, 2005). A few reasons exist to legitimize this issue, one reason is identified with the degree to which a client knows precisely what he or she is searching for and therefore his or her insight is considered the recovered results. Another critical reason is the presence of the alleged: spam web pages these are pages that from the search engine's perspective appear to be pertinent, yet, actually they contain no helpful data for clients (Gyongyi and Molina, 2005). In their discourse about web spam, Castillo *et al*. (2006) characterized web spam as any endeavor to swindle a search engine's pertinence calculation or an activity performed with the motivation behind impacting the positioning of the page. Distinguishing web spam is considered as a standout amongst the most difficult issues confronting search engines and web clients (Ntoulas *et al*., 2006). Since, the search engines are the doors to the World Wide Webs it is imperative to give the conceivable best results noting the client's inquiries. There are a few people surely understood as spammers attempt to deceive the search engines by boosting their web pages rank, thus, catch client regard for their pages. These pages contain a couple or not any helpful data that the client hopes to discover. The search engines need to distinguish or channel spam pages to give fantastic results to clients (i.e., really important pages). For a search engine to be assessed as an effective one it might return however much records as could reasonably be expected as well as

Corresponding Author: T. Muralidharan, Department of CSE, Dr. MGR Educational and Research Institute University, Maduravoyal, Chennai, India

ought to give back those important reports that are sans spam. At present, numerous procedures are connected via. search engines to battle spam, for example, distinguishing spam web pages through substance examination (Ntoulas *et al*., 2006). This method is the most prevalent procedure for spam discovery at present utilized via. search engines for example, Google in any case, it is still need to discover all spam web pages.

Spam can be exceptionally irritating with regards to web index for a few reasons. Initially, for the situation there are money related favorable circumstances from web index, the presence of spam pages may bring down the chance for honest to goodness (lawful) site pages to get the income that they may acquire without spam. Second, the search engine may give unessential results that clients don't expect and along these lines, a non-trifling bit of time may be spent on-line swimming through such undesirable pages. At long last the internet searcher may waste essential assets on spam pages this incorporate wasting system network bandwidth (Crawling), wasting CPU cycles (Processing) and wasting storage room (Indexing) (Ntoulas *et al*., 2006).

Microsoft researchers Ntoulas *et al*. (2006) demonstrate that some specific top level spaces will probably contain spam than others do for instance, biz (Business) has most prominent rate of spam with 70% of all pages being spam, .us space comes in second place with 35% spam pages. In addition, pages written in some specific dialects will probably be spam than those written in different dialects, for example, pages written in French are the destined to be spam with the rate of 25% of being spam. Spammers demonstrated their incredibleness to adjust to the diverse organizations accessible for web pages, a few spamming strategies utilized by spammers to impact the positioning page calculations of web search tools. Every one of these systems are considered trying for site page spam location calculation, particularly for Contents-based methodology. The two fundamental classifications of spammer procedures are: term spamming and link spamming (Ntoulas *et al*., 2006, Becchetti *et al*., 2008).

In term spamming, numerous systems that alter the substance of the page are connected. The substance incorporates: the record body, the title, meta labels in HTML header, stay writings connected with URLs and page URLs. The spammers can join their spontaneous substance (i.e., spam) to one or a greater amount of these substance bringing about another page that can pass the spam channel with no uncertainty of being legitimate. Among all term spamming strategies, the most well-known one is body spamming (Ntoulas *et al*., 2006) in which terms are incorporated into the record body, a case is to incorporate particular terms as "Free give cash", "free establishment", "Guarantee you!", "free see" and so on.

Literature review: Because of the critical part that site pages involve as means for supporting electronic business (e-Commerce), web pages get to be alluring focus for every single diverse sort of dealers and advertisers to promote their items available to be purchased, get-rich-on-the-fly plans (Sahami *et al*., 1998; Zhang *et al*., 2004) and to get data about explicit sites. In addition they turn into an alluring focus for spammers to install their spam content.

Early proposed approaches for spam filtering depended generally on manually constructed pattern matching rules that should be tuned to every client's message (Raja and Rajagopalan, 2014). That is they permit clients to hand-assemble a guideline set that comprises of an arrangement of coherent standards to recognize spam messages and web pages. Not with standing, these methodologies are appeared to be monotonous and tricky, since, clients need to give careful consideration just to assemble the sought arrangement of tenets which by the way not all clients can fabricate such a set.

Also, it is a period expending process, subsequent to the created set of standards ought to be changed or refined occasionally as the way of spam changes as well. On account of the issues connected with the manual development of principles, another methodology was proposed by Koprinska *et al*. (2007) to consequently adjust to the changing way of spam after some time and to give a framework that can gain straightforwardly from information as of now put away in the web server databases. These methodologies demonstrated as effective when connected for general order assignments, that is the characterization of e-Mail to either spam or non-spam taking into account their content, without any respects to the presence of some domain specific features.

A few machine learning algorithms have been proposed for content arrangement (order) (Su and Zhang, 2006; Yu and Xu, 2008). These methodologies were researched to be utilized for spam filtering since it is seen as a content order issue. By Sahami *et al*. (1998) they connected a machine learning algorithm with the end goal of spam filtering. In this algorithm, the channel figures out how to characterize archives into settled classes (i.e., spam and non-spam), taking into account their substance in the wake of being prepared on physically characterized reports.

Other than rule based methodologies discussed before, a lot of work was seen in the writing to consequently perform content-based classification. Bayes

classifiers (Zhang and Li, 2007) was proposed as a decent case of those methodologies that indicated attractive results with regards to e-Mail spam filtering.

Noi proposed a blend of graph neural network and probability mapping graph self organized maps composed into a layered design (Yates and Neto, 1999). It was a blend of unsupervised and supervised learning approaches yet the training time was computationally costly.

Erdelyi accomplished superior classification results in investigation utilizing learning techniques logit boost and random forest with less calculation hungry substance (Becchetti *et al*., 2008). They utilized 100,000 hosts from web spam UK 2007 and 190,000 hosts from DC 2010 datasets and examined the tradeoff between feature generation and spam classification accuracy. They demonstrated that more components enhance execution however complex elements, for example, PageRank enhances the characterization accuracy hardly.

MATERIALS AND METHODS

Hyrid Extensive Machine Learning Algorithm (HEMLA) using ANN Multi-Layer Perceptron (MLP): Hybrid extensive machine learning algorithm is a combination of many learning algorithms like conjugate gradient, resilient back-propagation and Levenberg-Marquardt algorithms. The ANN is an accumulation of basic preparing units which speak with each other utilizing a vast number of weighted associations. Every unit gets contribution from neighbor units or from outer source and processes yield which spreads to different neighbors. There is additionally a system to alter weights of the associations. Typically, there are 3 sorts of units:

- Input unit which gets signal from outer source
- Output unit which sends information out of the system
- Hidden unit which receives and sends signals inside the system

Huge numbers of the units can work parallel in the framework. ANN can be adjusted to take an arrangement of inputs and produce a desired set of outputs. This procedure is known as learning or training. There are two sorts of training in neural system:

- Supervised the network is given an arrangement of inputs and related output designs called training dataset to prepare the network
- Unsupervised the network trains itself by making bunches of examples. Here, no earlier set training information is given to the network

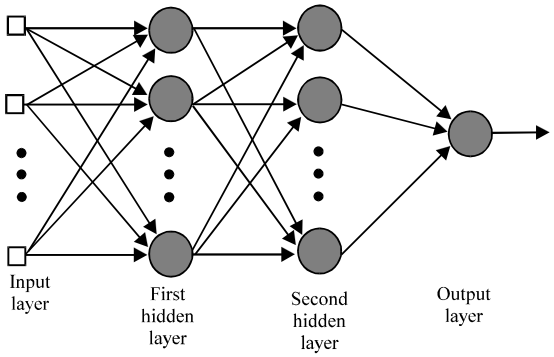

Fig. 1: A multilayer perceptron with one input, two hidden and one output layer

MLP is a nonlinear feed forward network model which maps a set of inputs x into a set of outputs y. It has 3 types of layers: input layer, output layer and hidden layer. Standard perceptron calculates a discontinuous function:

$$\vec{x} \to f_{step}(w_0 + (\vec{w}, \vec{x}) \quad (1)$$

Smoothing is done using a logistic function to get:

$$\vec{x} \to f_{log}(w_0 + (\vec{w}, \vec{x}) \quad (2)$$

Where:

$$f_{log}(z) = \frac{1}{1+e^{-z}}$$

MLP is a limited non-cyclic diagram where the nodes are neurons with logistic actuation. Neurons of ith layer serves as contribution for neurons of (i+1)th layer. Extremely mind boggling capacities can be computed with system containing numerous neurons. Every one of the neurons of one layer is associated with all neurons of next layer.

As inputs are bolstered into the input layer they get increased by interconnection weights while going from the input layer to the main hidden layer. Inside the first hidden layer they get summed and afterward handled by a nonlinear activation function. Every time data is prepared by a layer it gets increased by interconnection weights, then summed and handled by the following layer. At long last the data is processed one final time inside the output layer to create the neural network output (Fig. 1).

Supervised learning algorithms: While implementing any learning algorithm, our objective is to reduce the global error defined by E by:

$$E = \frac{1}{p}\sum_{p=1}^{p} E_p \quad (3)$$

Where:
P = The total size of training dataset
E_p = The error for training pattern p

And also:

$$E_p = \frac{1}{2}\sum_{i=1}^{N}(O_i - t_i)^2 \quad (4)$$

Where:
N = Total number of output nodes
O_i = Output of the ith output node and t_i is the target output at the ith output node

Every learning algorithm tries to reduce the global error by adjusting weights and biases. Now first we will discuss the three algorithms one by one.

Conjugate Gradient (CG) algorithm: It is fundamental back-propagation algorithm. It changes weights in the steepest plunge direction, i.e., the most negative of the gradients. This is the direction in which the capacity is diminishing generally quickly. It is watched that despite the fact that the capacity diminishes most quickly along the negative of the gradient direction it doesn't generally, give the speediest convergence. In the conjugate gradient calculation a pursuit is done in conjugate directions which for the most part gives the quicker joining than the steepest descent direction (Han and Kamber, 2000).

Conjugate gradient algorithm adjusts step size in each iteration along the conjugate gradient direction to minimize performance function. It searches the steepest descent direction on the first iteration:

$$p0 = -g0 \quad (5)$$

Then, it performs line search to determine the optimal distance to move along the current search direction by combining new steepest descent direction with the previous direction:

$$p_k = -g_k + \beta_k\, p_{k-1} \quad (6)$$

where, the constant β_k is:

$$beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \quad (7)$$

It is the ratio of the norm squared of current gradient to the norm squared of the previous Gradient (Koprinska *et al*., 2007).

Resilient Back-propagation (RB) algorithm: Multilayer perceptron networks ordinarily utilize sigmoid transfer function in the hidden layer. It is otherwise called squashing capacity since it packs an interminable information range into limited output range. The sigmoid functions slope goes nearer to zero as info gets extensive. It causes issue when we utilize steepest drop to prepare system. Since, the gradient may have little changes in values it might bring about little changes in weights and biases despite the fact that the weight and biases are genuinely far from the ideal qualities.

The Resilient Back-spread (RB) learning is useful in a way to evacuate the hurtful impact of the extent of incomplete subordinates. It utilizes just the indication of the partial derivative to decide the bearing of the weight upgrade. The size of the weight change is computed as taking after guideline (Lai, 2007):

- Estimations of every weight and bias are expanded when the derivative of performance function as for that weight has same sign for two cycles
- Values of every weight and bias are diminished when the derivative as for weight changes sign from the past emphasis
- If the derivative is zero the estimations of the weights and biases stay same
- When weights change then and there, estimations of weight change is diminished
- If weights constantly change in the same course, weight change is expanded

Levenberg-Marquardt (LM) algorithm: It provides a numerical solution to the problem of minimizing a generally non-linear function, over a space of parameters for the function. It is popular alternative to the Gauss-Newton method of finding the minimum of a function. It approaches second order training speed without computing the Hessian Matrix (Raja and Rajagopalan, 2014), if the performance function is of the form of a sum of squares then Hessian matrix can be approximated as:

$$H = J^T J \quad (8)$$

And the gradient is:

$$g = J^T e \quad (9)$$

Where:
J = The Jacobian matrix containing the first derivatives of network error with respect to weights and biases
e = The vector of network errors

The Jacobian matrix can be computed through a standard back-propagation technique that is much complex than computing Hessian matrix. This approximated matrix is used in following Newton like update:

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \quad (10)$$

At the point when the scalar, $\mu = 0$ it turns into the Newton's technique on approximated Hessian grid. At the point when μ is substantial it gets to be angle plunge with little step size. The Newton's strategy is quicker and more precise so calculation shifts towards it. So, the μ is diminished with each effective step (i.e., at the point when execution capacity diminishes). It is expanded when a conditional step would build the performance function. So, the performance function dependably lessens at every cycle. It is more intense than traditional gradient decent technique.

LM algorithm is very sensitive to initial network weights. It does not consider outliers in the data which may lead to over-fitting noise. To avoid these situations, regularization technique is used. One of such technique is Bayesian regularization.

Bayesian Regularization (BR): It is utilized to overcome the issue of inserting noise data. Mackay proposed Bayesian framework which can be specifically connected to the NN learning problem. It permits to evaluate compelling number of parameters really utilized by the model, i.e., the quantity of system weights really expected to take care of a specific issue. Bayesian regularization extends the cost capacity to hunt down negligible error as well as for insignificant error utilizing insignificant weights.

By utilizing Bayesian regularization one can maintain a strategic distance from immoderate cross approval. It is especially valuable for the issues that can't or would endure if a segment of accessible information were held to an approval set. Moreover, the regularization likewise decreases or wipes out the requirement for testing distinctive quantities of hidden neurons for an issue. Bayesian regularized ANN are hard to be over prepared and over-fit.

Algorithm evaluation: Confusion matrix or contingency table is utilized to assess the execution of a machine learning classifier. We utilized four attributes of Confusion matrix while assessing the execution of the calculations. These characteristics are sensitivity, specificity, efficiency and accuracy. These qualities are characterized as follows: Sensitivity or True Positive Rate (TPR), also known as recall rate is given by:

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (11)$$

Specificity or True Negative Rate (TNR) is given by:

$$\text{Specificity} = \frac{TN}{FP+TN} \quad (12)$$

Efficiency is given by:

$$\text{Efficiency} = \frac{\text{True Positive Rate} + \text{True Negative Rate}}{2} \quad (13)$$

Accuracy is given by:

$$\text{Accuracy} = \frac{TP+TN}{(P+N)} \quad (14)$$

Where:

P = Number of Positive instances
N = Number of Negative instances
TP = Number of Correctly classified positive instances
TN = Number of Correctly classified Negative instances
FP = Number of incorrectly classified as positive instances
FN = Number of incorrectly classified as Negative instances

RESULTS AND DISCUSSION

We evaluated 3 supervised learning algorithms which uses back-propagation on multilayer perceptron neural network. We checked the performance of these algorithms to automatically detect web spam. These algorithms are Conjugate Gradient, Resilient Back-propagation and Levenberg-Marquardt learning. We created neural network with single hidden layer and used one neuron in the output layer. We employed bipolar sigmoid function which has the output range of [-1, 1]:

$$f(x) = \frac{2}{1+e^{-ax}} - 1 \quad (15)$$

Stopping Criteria as Number of Iterations $\theta = 100$, learning rate $\alpha = 0.1$ number of neurons in hidden layers = 10 (or 20 where specified). A corpus is created with 368 instances of manually selected web pages in which about 30% instances were labelled as spam and rest of the pages were labelled as non-spam. We selected randomly about 80% of the training dataset from the corpus and for testing we used the remaining 20% of the records. We extracted total 31 low cost quality features of the pages and categorized them in 3 categories: URL (10 features), content (16 features) and Link (5 features). We call these factors low cost because they are computationally less expensive to be extracted. These features are listed.

URL features:

- SSL certificate
- Length of URL
- URL is not a sub-domain
- TLD is authoritative (*.gov, *.edu, *.ac)
- Domain contains more than 2 same consecutive alphabet)
- Sub-domain is more than level 3 (e.g., mocrosoft.com.my domain net)
- Domain contains many digits and special symbols
- IP address instead of domain name alexa top 500 site
- Domain length

Content features:

- HTML length
- Word count
- Text character length
- Number of images
- Description length
- Existence of H_2
- Existence of H_1
- Video integration
- Number of ads
- Title character length
- Compression ratio of text
- Ratio of text to HTML
- Presence of alt text for images
- Presence of obfuscated JavaScript code (pop-ups, redirections etc.)
- Percentage of call to action in the text
- Percentage of stop words in text

Links features:

- Number of internal links
- Internal link is self referential
- Number of external links
- Fraction of anchor text/total text
- Word count in anchor text

To test the performance of each algorithm, we trained, tested and obtained 10 performance result values for each category and calculated the average. Tables 1-6 show the performance results of each algorithm. Consider the number of neurons in hidden layer as 10 unless specified. The values in the tables represent average of 10 experimental readings of each category. The values in underlined show the best results. Data from the tables suggest that Conjugate Gradient (CG) algorithm gave best TPR (Sensitivity) in most of the categories, whereas, Levenberg-Marquardt algorithm with Bayesian Regularization (LM+BR) gave best TNR (Specificity) in most of the categories. The general best grouping execution was accomplished in the majority of the classes by Resilient Back-propagation (RB) calculation in both efficiency and accuracy measures. The training time was discovered least in the majority of the cases for Resilient Back-propagation (RB), so, we can say that it is not just best in arrangement it is the fastest algorithm too.

Table 1: Using only URL features (10 features)

Algorithm	Sensitivity (TPR)	Specificity (TNR)	Efficiency	Accuracy	Training time (sec)
CG	0.9153	0.3088	0.6121	0.5716	0.149
RB	0.4653	0.9117	0.6885	0.7183	0.168
LM	0.6884	0.7647	0.7265	0.7316	2.723
LM+BR	0.4224	0.9264	0.6744	0.7090	5.790

Table 2: Using only content features (16 features)

Algorithm	Sensitivity (TPR)	Specificity (TNR)	Efficiency	Accuracy	Training time (sec)
CG	0.7853	0.8232	0.8104	0.8200	1.265
RB	0.7265	0.9244	0.8237	0.8336	0.196
LM	0.7114	0.8412	0.7524	0.7628	8.354
LM+BR	0.6224	0.9324	0.7748	0.7956	13.350

Table 3: Using only links features (5 features)

Algorithm	Sensitivity (TPR)	Specificity (TNR)	Efficiency	Accuracy	Training time (sec)
CG	0.9515	0.6384	0.7845	0.7616	0.145
RB	0.6654	0.9144	0.7984	0.8045	0.168
LM	0.7254	0.9045	0.8145	0.8121	2.354
LM+BR	0.7028	0.8844	0.7984	0.8041	3.054

Table 4: Using URL and links features (15 features)

Algorithm	Sensitivity (TPR)	Specificity (TNR)	Efficiency	Accuracy	Training time (sec)
CG	0.9548	0.7898	0.8845	0.8678	1.024
RB	0.8547	0.9844	0.8904	0.9145	1.895
LM	0.8151	0.9355	0.8447	0.8841	3.587
LM+BR	0.8929	0.9589	0.9984	0.9156	8.925

Table 5: Using URL+content features (26 features)

Algorithm	Sensitivity (TPR)	Specificity (TNR)	Efficiency	Accuracy	Training time (sec)
No. of neurons in hidden layer = 10					
CG	0.8845	0.9267	0.9054	0.9178	0.3024
RB	0.8527	0.9765	0.9042	0.9415	0.2189
LM	0.8251	0.9886	0.8845	0.8818	9.5471
LM+BR	0.7989	0.9879	0.8545	0.8872	22.4580
No. of neurons in hidden layer = 20					
CG	0.8548	0.9558	0.985	0.9048	0.687
RB	0.8797	0.9823	0.9204	0.9245	0.467
LM	0.8871	0.9255	0.7647	0.9041	28.878
LM+BR	0.7929	1.4809	0.8894	1.9968	218.245

Table 6: Using URL+content features (31 features)

Algorithm	Sensitivity (TPR)	Specificity (TNR)	Efficiency	Accuracy	Training time (sec)
No. of neurons in hidden layer = 10					
CG	0.8545	0.9367	0.9124	0.9168	0.2824
RB	0.8727	0.9739	0.9284	0.9325	0.2891
LM	0.8651	0.9024	0.8816	0.8478	11.4710
LM+BR	0.6989	0.9819	0.8345	0.8572	78.8870
No. of neurons in hidden layer = 20					
CG	0.8481	0.9856	0.9101	0.9218	0.578
RB	0.8870	0.9544	0.9144	0.9251	0.3647
LM	0.8645	0.9285	0.8617	0.9001	48.548
LM+BR	0.8119	9.9988	0.8994	0.9128	289.115

Conjugate Gradient (CG) works quicker when number of sources of info is less, yet when number of data sources is expanded, it turns out to be slower than RB. The slowest calculation watched was Levenberg-Marquardt with Bayesian Regularization (LM+BR) in the majority of the cases in the investigation.

The overall classification performance of every calculation enhanced with expanded number of page quality components, yet, preparing time additionally expanded. The situations where the quantity of value components were high, expanding number of neurons enhanced the grouping execution of algorithms however it is likewise expanded the training time.

CONCLUSION

Detecting spam web pages is one of the major challenges that face search engines in their queries results. Search engines should return high quality results in response to the user's queries. Many search engines necessitate an integration of a healthy detection spam to eliminate all web pages that effect in page ranking algorithm. Several content based and machine learning techniques were proposed to detect spam pages.

The researcher involved hybrid extensive machine learning algorithms discovering non required pages. The experimental results showed that resilient back propagation algorithm is fastest and performs best in both efficiency and accuracy measures. Conjugate Gradient algorithm gives best sensitivity whereas Levenberg Marquardt algorithm with Bayesian Regularization gives best specificity but it is the slowest when training time is considered.

Classification performance of each algorithm improves with increased input factors. If the number of factors is high increased number of neurons improves the performance of algorithm. The training time increases for all algorithms when either number of inputs are increased or number of neurons in hidden layer are increased.

RECOMMENDATION

For future researches, we will develop an application as plug-in in one of the open source browser to work as detector in online website pages to notify the users for spam pages currently working on.

REFERENCES

Becchetti, L., C. Castillo, D. Donato, R.B. Yates and S. Leonardi, 2008. Link analysis for web spam detection. ACM. Trans. Web, Vol. 2, 10.1145/1326561.1326563.

Castillo, C., D. Donato, L. Becchetti, P. Boldi, S. Leonardi and M. Santini *et al*., 2006. A reference collection for web spam. ACM. Sigir Forum, 40: 11-24.

Gyongyi, Z. and H.G. Molina, 2005. Web spam taxonomy. Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005), May 10-14, 2005, Stanford University, Chiba, Japan, pp: 1-9.

Han, J. and M. Kamber, 2000. Data Mining: Concept and Techniques. Morgan Kaufmann Publishers, New York, USA.,.

Koprinska, I., J. Poon, J. Clark and J. Chan, 2007. Learning to classify E-mail. Inf. Sci., 177: 2167-2187.

Lai, C.C., 2007. An empirical study of three machine learning methods for spam filtering. Knowledge-Based Syst., 20: 249-254.

Ntoulas, A., M. Najork, M. Manasse and D. Fetterly, 2006. Detecting spam web pages through content analysis. Proceedings of the 15th International Conference on World Wide Web, May 23-26, 2006, ACM, Edinburgh, Scotland, ISBN:1-59593-323-9, pp: 83-92.

Raja, V.S. and S.P. Rajagopalan, 2014. A comparative analysis of optimization techniques for artificial neural network in bio medical applications. J. Comput. Sci., 10: 106-106.

Sahami, M., S. Dumais, D. Heckerman and E. Horvitz, 1998. A bayesian approach to filtering junk E-Mail. Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, July 26-27, 1998, Madison, Winconsin, pp: 55-62.

Su, J. and H. Zhang, 2006. Full Bayesian network classifiers. Proceedings of the 23rd International Conference on Machine Learning, June 25-29, 2006, ACM, Pittsburgh, Pennsylvania, ISBN:1-59593-383-2, pp: 897-904.

Yates, R.B. and B.R. Neto, 1999. Modern Information Retrieval. Pearson Education, Upper Saddle River, New Jersey, ISBN:978-81-317-0977-1, Pages: 517.

Yu, B. and Z.B. Xu, 2008. A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowl. Based Syst., 21: 355-362.

Zhang, H. and D. Li, 2007. Naive bayes text classifier. Proceedings of the 2007 IEEE International Conference on Granular Computing (GRC 2007), November 2-4, 2007, IEEE, Nova Scotia, Canada, ISBN:0-7695-3032-X, pp: 708-708.

Zhang, L., J. Zhou and T. Yao, 2004. An evaluation of statistical spam filtering techniques. ACM Trans. Asia Language Inform. Proc., 3: 243-269.