International Journal of Computer Networks and Communications Security

VOL. 3, NO. 4, APRIL 2015, 191–199
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Web Spam Detection Inspired by the Immune System

MAHDIEH DANANDEH OSKOUEI[1] and SEYED NASER RAZAVI[2]

[1] Department of Computer, Shabestar Branch, Islamic Azad University, Shabestar, Iran

[2] Department of Electrical and Computer Engineering, University of Tabriz, Iran

E-mail: [1]r mah.danandeh@gmail.com, 2 n.razavi@tabrizu.ac.ir

ABSTRACT

Internet is a global information system, and search engines are currently the most common tools used to find information in web receiving query from the user, and present a list of the results related to user query. Web spam is an illegal way to increase web pages rank, and it tries to increase the rank of some web pages in the list of results by manipulating ranking algorithm of search engines. In this paper, a novel method is presented to detect spam content on the web. It is based on classification and employs an idea from biology, namely, danger theory, to guide the use of different classifiers. The evaluation of content features of WEBSPAM-UK2007 data set using 10-fold cross-validation demonstrates that this method provides high evaluation criteria in detecting web spam.

Keywords: *Artificial immune system, Web spam, Danger theory, Machine learning, Classification.*

1 INTRODUCTION

Artificial immune system is relatively a new science, and has been derived from the performance of body immune system when it encounters with pathogens. With regard to performance and complex defense mechanisms of natural immune system in living organisms against pathogens, researchers have designed artificial immune system by simulating this system, so that they can solve engineering problems. The research diversity created by using the method of artificial immune system indicates the ability to solve complex engineering problems thorough using algorithms presented in terms of artificial immune system. Also, it has provided an interesting research background in various fields.

Web spam has been considered as one of the common problems in search engines, and it has been proposed when search engines appeared for the first time. The aim of web spam is to change the page rank in query results. In this way, it is placed in a rank higher than normal conditions, and it is preferably placed among 10 top sites of query results in various queries.

Web spam was recognized a spamdxing (a combination of spam and indexing) for the first time, and later search engines tried to combat with this difficulty [1]. With regard to the paper presented by Davidson in terms of using machine learning for web spam detection, this topic has been considered as an university discussion [2]. Since 2005, AIRWeb workshops have considered some places where the researchers interested in web spam exchange their opinions [1]. Web spam is the result of using illegal and immoral methods to manipulate web result [3-5]. According to definition presented by Gyongyi and Garcia, web spam refers to an activity performed by some people to change the rank of a web page illegally [4]. Wu et al. have introduced web spam as a behavior that deceives search engines [6]. Web spam has been considered as a challenge in search engines [7]. It reduces not only the quality of search engines but also the trust of users and search engine providers. Also, it wastes computing resources of search engines [8]. If an effective solution is presented to detect it, then search results will be improved, and users will be satisfied in this way.

One of the theories that has been proposed by Matzinger in terms of immunology is danger theory [9, 10]. This theory has been recently used in artificial immune system. We have considered danger theory to detect web spam by using web pages classification. The new proposed method has investigated its performance in content features of

WEBSPAM-UK2007 data set. Also, we have compared this method with popular ensemble classification methods. The results show that method based on danger theory can improve classification of web spam pages. The rest of this paper has been organized as follows. In section II, we have presented related studies in terms of web spam detection, and the main concepts of danger theory have been explained. It also reviews used classifications methods. In section III, the framework of our proposed method and the way of using danger theory concepts in machine learning have been proposed. In section IV, the results of evaluation have been described, and finally, in section V, conclusion and the future work have been presented.

2 LITERATURE REVIEW AND MAIN CONCEPTS

This section reviews some of the most important works in the past devoted to web spam detection using machine learning techniques, used classifiers in our method and three ensemble classifiers. It also contains fundamental concepts related to danger theory.

2.1 Web spam detection by machine learning techniques

Ntoulas et al. [11] took into account detection of web spam through content analysis. Amitay et al. [12] have considered categorization algorithms to detect the capabilities of a website. They identified 31 clusters that were a group of web spam.

Prieto et al. [13] presented a system called SAAD in which web content is used to detect web spam. In this method, C4.5, Boosting and Bagging have been used for classification. Karimpour et al. [14] firstly reduced the number of samples, and then they considered semi-supervised classification method of EM-Naive Bayesian to detect web spam. Rungsawang et al. [15] applied ant colony algorithm to classify web spam. The results showed that this method, in comparison with SVM and decision tree, involves higher precision and lower Fall-out. Silva et al. [16] considered various methods of classification involving decision tree, SVM, KNN, LogitBoost, Bagging, adaBoost in their analysis.

Becchetti et al. [17] considered link characterization such as TrustRank and PageRank to classify web spam. Castillo et al. [18] took into account link-based characterization and content analysis by using C4.5 classifier to classify web spam. Dai et al. [19] classified temporal features through using two levels of classification. The first level involves several SVM^{light}, and the second level involves a logistic regression.

In this paper, we used danger theory for the first time to provide a combined method in order to detect web spam. In our method, spam pages are classified on the basis of high precision and Accuracy.

2.2 Main concepts of danger theory

One of the theories proposed in immunology is danger theory suggested by Anderson and Matzinger. According to this theory, immune systems response to the stimulation whom body recognizes as a harmful element. Hence, the main concept of this theory is to response to the danger to which immune system responses instead of responding to nonself [20]. According to danger theory, both immune and nonself cells are observed together, and when there is an attack, those cells that die unnaturally create a signal called danger signal before their death [21]. They are distributed in a small zone around the cell. This zone is called danger zone. Immune system is activated just in this zone, and it responses to antigens of this zone. There is another vision in danger theory, and it's a model containing two signals. This model has been suggested by Bretscher and Cohn [22]. There are two signals in this model:

- The first signal is antigen detection.
- The second signal is stimulation aid.

In artificial immune system, there are different definitions for danger signal, and this depends on the type of problem. Danger has been considered as a false behavior in irregular detection systems, and it has been taken into account as a false data in fraud detection systems. Also, it has been considered as an attractive data in data mining [23].

Since danger theory is a new theory, there are limited papers in this regard. The first use of this theory to detect self is nonself. If detecting self from nonself is required, then using danger theory will be effective [24]. Aickelin has presented a system for intrusion detection on the basis of danger theory [25]. Secker, in hid PhD thesis, has proposed two artificial immune systems involving a system called AISEC for classification of emails and another system is called AISIID to search web pages. Both systems have been created on the basis of danger theory. In AISEC, danger signal is created on the basis of user reaction to classified emails [26]. Zhu et al. [27] presented a method on the basis of danger theory to detect spam.

The following points should be taken into account when danger theory is used:

- Danger depends on the application, and signal may not be related to danger.
- Signal may be positive or negative.
- Danger zone is an area in biology where it can be replaced by another criterion in artificial immune system such as temporal features and etc.

2.3 The classifiers used in the proposed method and comparisons

To evaluate our method, we have used the following seven different classifiers: NN, C4.5, Bayes network, Random forest, Single layer perceptron, AIRS2 and Random Tree. We compared the evaluation results of each combination with base classifiers results. Also, in order to be sure of results optimization, we compared evaluation criteria with the result of three ensemble classifiers involving Vote, Stacking and Grading. In the following subsections, these classifiers will be explained.

2.3.1 KNN

KNN is supervised learning algorithm, and it is related to sample based learning methods used to estimate objective function with discrete or continuous values. In this classification method, a sample is classified according to k samples whose features and characteristics have the most similarity with that sample. In order to apply this method, criteria like Euclidean distance and Manhattan distance are firstly considered to measure the similarity. The distance of new sample is computed according to training data, and k samples having the nearest distance to the new sample are detected. In this k samples, the number of each class member is identified, and the new sample is classified according to the label of the class having more repetition.

2.3.2 Decision tree (C4.5)

Decision tree is one of the supervised learning algorithms, and it is widely used in machine learning problems due to its simplicity and efficiency. In this method, the model is implemented on the basis of a tree. In this method, a tree is created on the basis of a training set, and according to this tree, the new member is classified in a special class. In this way, searching initiates from the root of tree, and after browsing middle nodes, it ultimately reaches the levels. Internal nodes identify the features. These features address a question about input example. In each internal node, there is a branch on the basis of the number of possible answers for this question and each one is identified with the value of that answer. The levels of this tree are identified with a class. There are various algorithms to create the trees and tree pruning. Often, learning algorithms of decision tree perform on the basis of a greedy search method in top- down way in available trees. One of learning algorithms is C4.5 algorithm that is a developed version of ID3. When a tree is created for each node, the best feature is selected by using a criteria based on entropy. After creating the tree, pruning method is used, and the smallest tree is obtained [28].

2.3.3 Bayes network

Classification algorithm of Bayes network is a supervised learning algorithm based on Bayes theory. In this method, conditional probability is computed for each node, and a Bayes network is formed. Bayes net is a directed acyclic graph, and it has been created from a set of nodes and edges. In this classification method, variable kinds of algorithms are used to estimate conditional probability such as simple estimator and Bayes net estimator. Various search algorithms used for tree structure learning are genetic algorithm, hill climber algorithm and K2 [29].

2.3.4 AIRS2

AIRS2 is a supervised learning algorithm based on artificial immune system. Immune mechanisms that have been used in this classification algorithm involve clonal selection, affinity maturation and affinity recognition balls (ARBs) [30].

2.3.5 Random forest

One of the classifiers using Bagging method is random forest involving several decision trees, and their output is obtained from the output of individual trees. This algorithm synthesizes Bagging method by random selection of features so that diverse decision trees can be created. One of its advantages is high number of input.

2.3.6 Single layer perceptron

The simplest form of using perceptron is to use them in a single layer. A single layer perceptron involves a number of input nodes connected to a number of perceptron located in a layer (output

layer). Attributes are given a weight multiplied by the value of the attribute, and then are summed up to find the output. Each weight is given an initial arbitrary value, and the error is calculated. Then, the model incrementally adjusts the weights to reduce the error. After much iteration, the model is able to predict the output accurately.

2.3.7 Random tree

It is a kind of tree created randomly from a set of possible trees with k random feature in each node. Each tree is equal in a set of trees having sampling chance. In other words, trees distribution is uniform. Random trees can be effectively created, and a combination of random trees set results in presenting a precise model.

2.3.8 Vote

Vote is an ensemble classification method since it does not use meta-classifier, it is similar to MultiScheme. There are various vote designs. However, vote uses a simple combination design of the main classifiers predictions for ensemble prediction.

2.3.9 Stacking

In this classifier, the predictions of main classifiers are used as a feature in new training dataset, and main class labels are kept. This new training dataset is learnt by using a meta-classifier to obtain ensemble prediction.

2.3.10 Grading

Grading is an ensemble meta-classifier method presented by Seewald and Furnkranz. This method has an opposite process to stacking method. In this method, the outputs of main classifiers are graded as false or true labels. Then, these graded results are combined.

3 WEB SPAM DETECTION BY USING THE METHOD OF MACHINE LEARNING BASED ON DANGER THEORY

In this section, a method is presented for classification based on danger theory. In immune system, danger is differentiated from nondanger on the basis of danger theory, so it can be used in classification problems involving two classes. We used danger theory involving two signals to present a combined classification method. The concepts of signal 1 and signal 2 have been explained in this method as follows:

- Signal 1 is detection of web spam pages which is produced by classifier1 for each test sample. If spam page is detected, then classifier 1 produces positive signal; otherwise, negative signal is produced, and this signal is only sent to a sample of test.
- Signal 2 is an auxiliary signal which is produced by classifier 2 for every test sample. If classifier 2 detects a web spam, it produces a positive signal; otherwise it produces a negative signal. For each test samples, signal 2 has been received from test samples which are located in the danger zone.

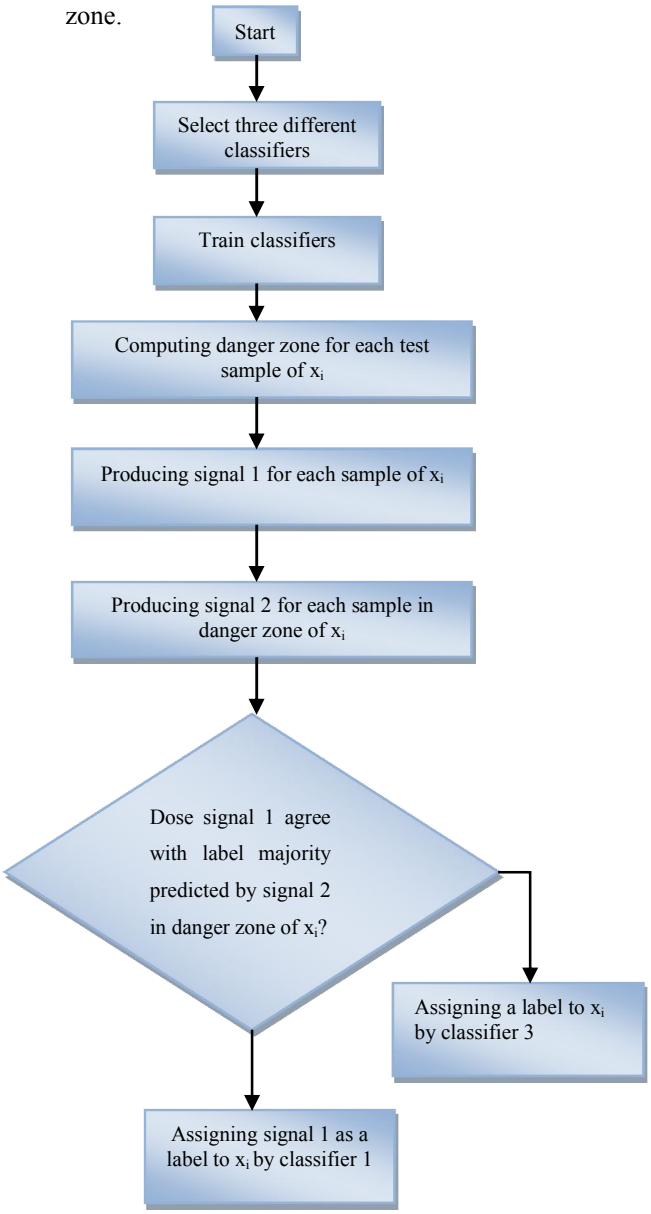

Fig. 1. Process of the proposed method.

Process of the proposed method has been shown in Figure 1, and its details are as follows:
Step1: Three various classifications are selected.
Step2: Classifiers are trained on dataset.
Step3: For each sample x_i in test set, danger zone is calculated. In this way, at first, Euclidean distance of x_i is calculated between x_i test sample and other test samples. Then, average is calculated according to obtained distances. θ is the size of danger zone of x_i and $||x_i - x_j||$ is Euclidean distance between x_i test sample and other test samples . If the considered sample is shown with a feature vector; that is, $\langle a_1(x), a_2(x), \ldots a_n(x)\rangle$, then Euclidean distance between two samples x_i and x_j is defined in formula (1).

$$||x_i - x_j|| = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \quad (1)$$

The average of Euclidean distances is defined in formula (2).

$$\theta = \underset{j=1}{\overset{n}{AVG}} ||x_i - x_j|| = \frac{\sum_{j=1}^{n} ||x_i - x_j||}{n} \quad (2)$$

Step 4: for each sample x_i, signal 1 is produced. This signal is sent to test sample x_i. In other words, classifier 1 assigns a lable for each sample x_i.
Step 5: for each test sample in danger zone of x_i test sample ($||x_i - x_j|| <= \theta$), signal 2 is produced and sent to test sample x_i. In other words, classifier 2 assigns a label to the test sample available in danger zone of x_i. This lable is used in final lable decision making for x_i.
Step 6: Voting is performed in predicted lables obtained in stage 5. If majority lable of samples in danger zone of x_i agree with x_i lable predicted in stage 3, then the lable predicted by classifier 1 is assigned as final lable; otherwise classifier 3 predicts the lable of x_i test sample.

4 RESULTS EVALUATION

To evaluate performance of the proposed method, this section uses WEBSPAM-UK2007 data set in order to compare the proposed method with different classification methods.

4.1 Data set

In this part, data set of WEBSPAM-UK2007 has been used to compute evaluation criteria. This is a general accessible data set used in web spam, and involves a huge set of spam /nonspam labeled by a group of volunteers collected from UK in May 2007. Dataset has been divided in two files: train set involving a small number of labeled hosts, and test set involving a large set of hosts without label. In this paper, we used train set for evaluation. This set involves 3849 hosts and features based on content and link. We have considered content analysis in our method, and it involves 96 features. Features employed in this paper are listed in Appendix I.

4.2 Evaluation criteria

In order to measure the classification performance of spam pages, we used the following criteria: precision, accuracy and FP rate.
Precision: it is the proportion of sample numbers that have been truly detected as positive (spam pages) to the total number of samples that have been detected as positive.

$$\text{Precision} = \frac{\text{True positive}}{\text{\# Predicted Positive}} = \frac{\text{True positive}}{\text{True Positive + False Positive}} \quad (3)$$

Accuracy: Accuracy refers to the proportion of samples that have been accurately classified to total number of samples.

$$\text{Accuracy} = \frac{\text{True Positive+True Negative}}{\text{True Positive + False Positive + True Negative + False Negative}} \quad (4)$$

FP rate:

$$\text{FP rate} = \frac{\text{False Positive}}{\text{False Positive+True Negative}} \quad (5)$$

4.3 Comparing results

In this paper, we have used 10- fold cross-validation method. In order to evaluate the proposed method, the results of 5 combinations have been presented in table 1. In table 2, the obtained result from Vote, Staking and Grading have been presented by using the same combinations used in the proposed method. In table 3, the obtained result from base classification methods used in the proposed method have been shown. Tables 1-3 are listed in Appendix II.

Figures 2-7 show the comparison of our method with used base classifiers and three ensemble classifiers of Vote, Staking and Grading.

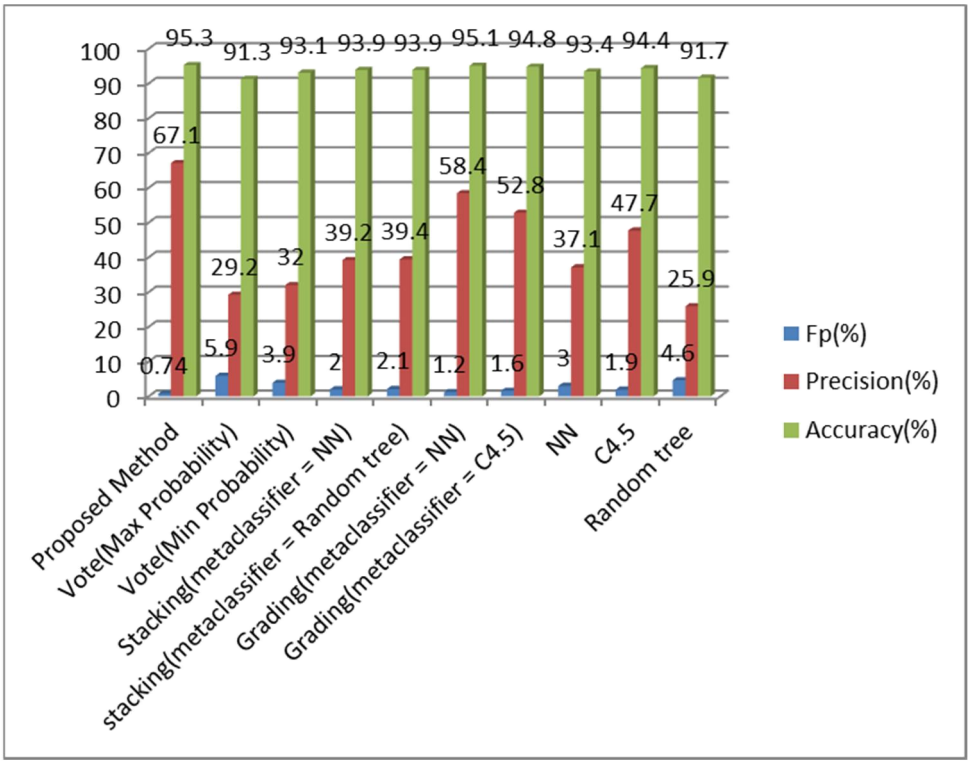

Fig. 2. Comparing the combination of NN, Random tree, C4.5 in the proposed method with other methods.

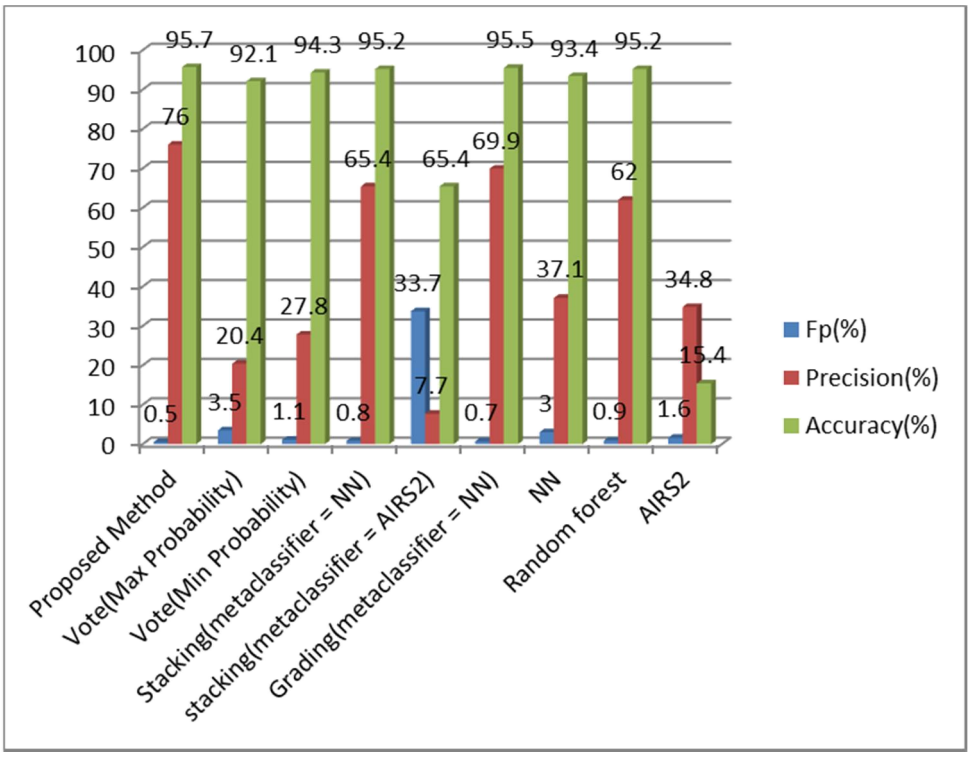

Fig. 3. Comparing the combination of NN, AIRS2, Random forest in the proposed method with other methods.

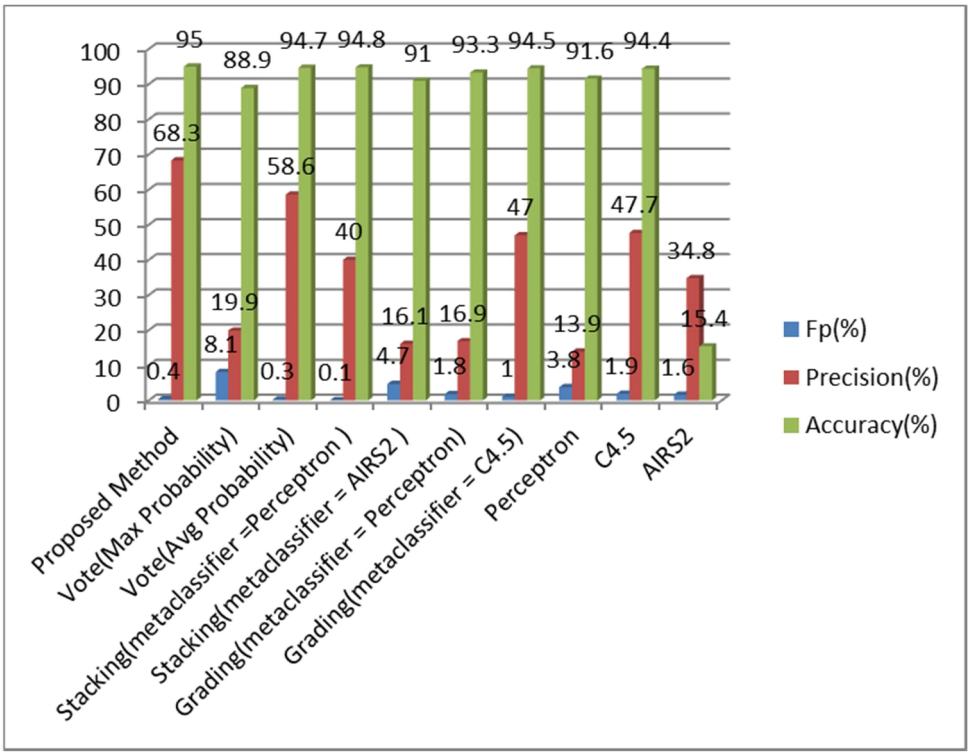

Fig. 4. Comparing the combination of Perceptron, C4.5, AIRS2 in the proposed method with other methods.

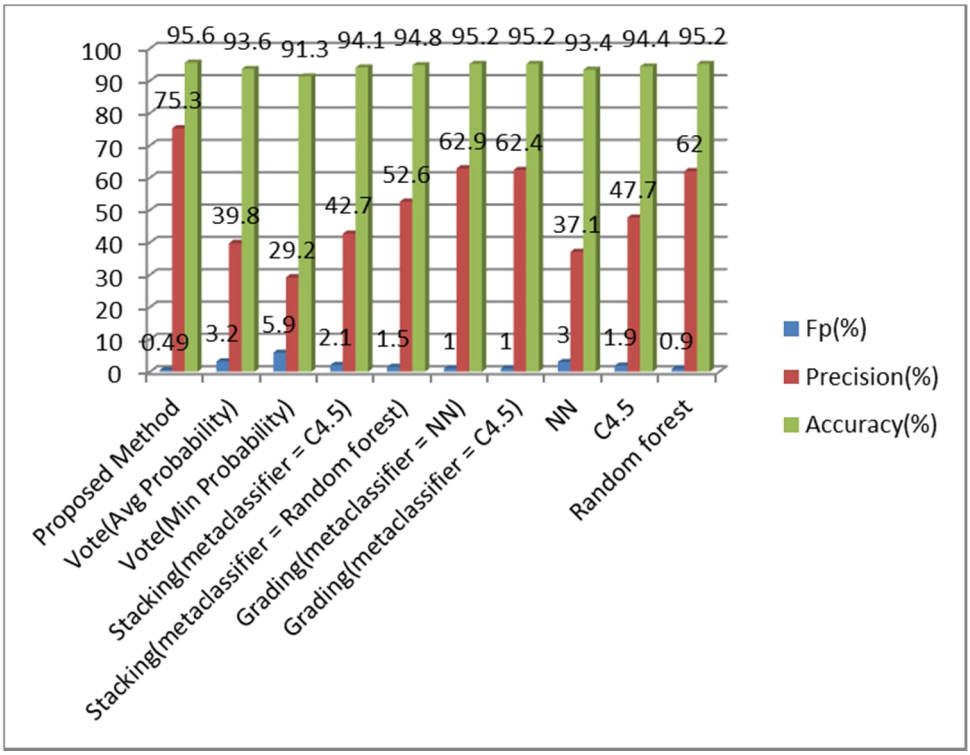

Fig. 5. Comparing the combination of NN, Random forest, C4.5 in the proposed method with other methods

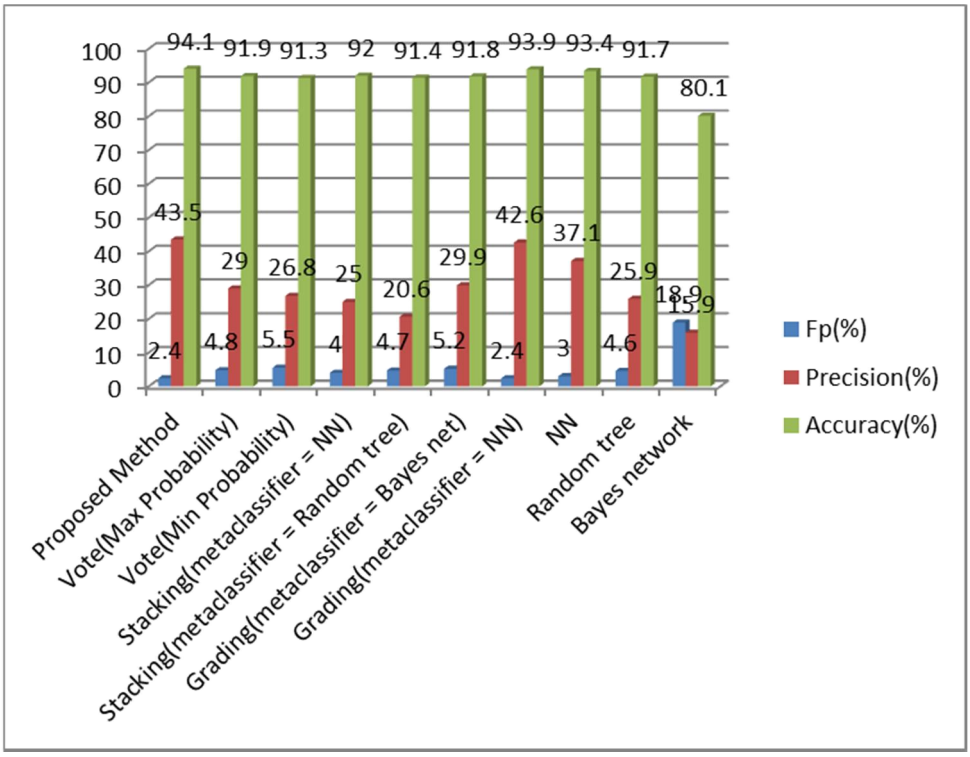

Fig. 6. Comparing the combination of Bayes network, Random tree, NN in the proposed method with other methods

In experiments, the order of classifiers in combinations of the propose method have been consider in the form of classifier 1, classifier 2, classifier 3. In Figure 2, the combination of NN, Random tree, C4.5 in the proposed method is compared to other classification methods. As this figure shows, this combination with values of FP= 0.74%, precision=67.1% and accuracy= 95.3% has better results than other classification methods shown in figure 2. Figure 3 shows the comparison of the combination of NN, AIRS2, Random forest in the proposed method and other classifiers. This combination with the values of FP=0.5%, precision= 76% and accuracy=95.7% has better results in comparison to other classification methods shown in this figure. Figure 4 shows the comparison of the proposed method with the combination of Perceptron, C4.5, AIRS2 and other classifiers. This combination with the values of FP=0.4%, precision=68.3%, accuracy=95% has optimal results in comparison to other classification methods investigated in figure. As it can be observed in figure 5, the combination of NN, Random forest, C4.5 with the values of FP=0.49%, precision=75.3% and accuracy=95.6% has optimal results in comparison to other classification methods investigated in figure. Figure 6 shows the comparison results of the proposed method with the combination of Bayes net, Random tree, NN in comparison to other classification methods. This combination with the values of FP=2.4%, precision=43.5% and accuracy=94.1% has better results than other classification methods investigated in the figure.

According to the results of tables 1-3, our method with the combination of NN, AIRS2, Random forest obtained the highest accuracy and precision among all methods with 95.7 and 76 success rates, respectively. It also can be observed that in the proposed method, the combination of Perceptron, C4.5, AIRS2 has best FP rate among all methods with 0.4% FP rate.

These results show that our proposed method is more effective in classification of spam pages.

5 CONCLUSION

In this paper, a combined method inspired from danger theory has been presented to detect web spam. The performance of the proposed method is evaluated using WEBSPAM-UK2007 data set. The results suggest a better performance for the proposed method in many cases when compared to the results obtained from base classifications methods and ensemble classifiers.

In spite the fact that existing methods for creating web spam evolves as time goes, it is possible to detect these types of pages by focusing on their features and using machine learning techniques. In

future, we will use a combination of danger theory and one of the algorithms of artificial immune system to detect web spam.

7 REFERENCES

[1] Najork M. Web Spam Detection. Encyclopedia of Database Systems. 2009;1:3520-3.

[2] Davison BD. Recognizing nepotistic links on the web. Artificial Intelligence for Web Search. 2000:23-8.

[3] Collins G. Latest search engine spam techniques. Aug 2004.

[4] Gyongyi Z, Garcia-Molina H. Web Spam Taxonomy. First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005). Chiba, Japan2005.

[5] Perkins A. The classification of search engine spam. 2001.

[6] Wu B, Goel V, Davison BD. Topical trustrank: Using topicality to combat web spam. Proceedings of the 15th international conference on World Wide Web: ACM; 2006. p. 63-72.

[7] Henzinger M, Motwani R, Silverstein C. Challenges in web search engines. SIGIR Forum. 2002;36:11-22.

[8] Abernethy J, Chapelle O, Castillo C. WITCH: A New Approach to Web Spam Detection. In Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb}2008.

[9] Matzinger P. The real function of the immune system or tolerance and the four d's (danger, death, destruction and distress). American Society for Microbiology. 1996.

[10] Matzinger P. The danger model: a renewed sense of self. Science. 2002;296:301-5.

[11] Ntoulas A, Najork M, Manasse M, Fetterly M. Detecting spam web pages through content analysis. the 15th International World Wide Web Conference. Edinburgh, ScotlandMay 2006. p. 83–92.

[12] Amitay E, Carmel D, Darlow A, Lempel R, Soffer A. The connectivity sonar: Detecting site functionality by structural patterns. the 14th ACM Conference on Hypertext and Hypermedia. Nottingham, UKAug 2003. p. 38–47.

[13] Prieto V, Álvarez M, López-García R, Cacheda F. Analysis and Detection of Web Spam by Means of Web Content. In: Salampasis M, Larsen B, editors. Multidisciplinary Information Retrieval: Springer Berlin Heidelberg; 2012. p. 43-57.

[14] Karimpour J, Noroozi A, Alizadeh S. Web Spam Detection by Learning from Small Labeled Samples. International Journal of Computer Applications. 2012;50:1-5.

[15] Rungsawang A, Taweesiriwate A, Manaskasemsak B. Spam Host Detection Using Ant Colony Optimization. In: Park JJ, Arabnia H, Chang H-B, Shon T, editors. IT Convergence and Services: Springer Netherlands; 2011. p. 13-21.

[16] Silva RM, Yamakami A, Alimeida TA. An Analysis of Machine Learning Methods for Spam Host Detection. 11th International Conference on Machine Learning and Applications (ICMLA)2012.

[17] Becchetti L, Castillo C, Donato D, Leonardi S, Baeza-Yates RA. Link-Based Characterization and Detection of Web Spam. AIRWeb 2006. Seattle, Washington, USA2006. p. 1-8.

[18] Castillo C, Donato D, Gionis A, Murdock V, Silvestri F. Know your neighbors: web spam detection using the web topology. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. Amsterdam, The Netherlands: ACM; 2007. p. 423-30.

[19] Dai N, Davison BD, Qi X. Looking into the past to better classify web spam. Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web. Madrid, Spain: ACM; 2009. p. 1-8.

[20] Anderson CC, Matzinger P. Danger: the view from the bottom of the cliff. Seminars in immunology: Academic Press; 2000. p. 231-8.

[21] Gallucci S, Matzinger P. Danger signals: SOS to the immune system. Current opinion in immunology. 2001;13:114-9.

[22] Bretscher P, Cohn M. A Theory of Self-Nonself Discrimination Paralysis and induction involve the recognition of one and two determinants on an antigen, respectively. Science. 1970;169:1042-9.

[23] Aickelin U, Cayzer S. The danger theory and its application to artificial immune systems. arXiv preprint arXiv:08013549. 2008.

[24] Aickelin U, Bentley P, Cayzer S, Kim J, McLeod J. Danger theory: The link between AIS and IDS? Artificial Immune Systems: Springer; 2003. p. 147-55.

[25] Aickelin U, Greensmith J. Sensing danger: Innate immunology for intrusion detection. Information Security Technical Report. 2007;12:218-27.

[26] Secker A. Artificial Immune Systems for Web Content Mining: Focusing on the Discovery of

Interesting Information: University of Kent; 2006.

[27] Zhu Y, Tan Y. A Danger Theory Inspired Learning Model and Its Application to Spam Detection. In: Tan Y, Shi Y, Chai Y, Wang G, editors. Advances in Swarm Intelligence: Springer Berlin Heidelberg; 2011. p. 382-9.

[28] Quinlan JR. C4. 5: programs for machine learning: Morgan kaufmann; 1993.

[29] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Machine learning. 1997;29:131-63.

[30] Watkins A, Timmis J, Boggess L. Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm. Genetic Programming and Evolvable Machines. 2004;5:291-317.

Appendix I

List of content features in WEBSPAMUK2007 Dataset.

HST_1
Number of words in the page (home page = hp)
HST_2
Number of words in the title (hp)
HST_3
Average word length (hp)
HST_4
Fraction of anchor text (hp)
HST_5
Fraction of visible text (hp)
HST_6
Compression rate of the hp
HST_7
Top 100 corpus precision (hp)
HST_8
Top 200 corpus precision (hp)
HST_9
Top 500 corpus precision (hp)
HST_10
Top 1000 corpus precision (hp)
HST_11
Top 100 corpus recall (hp)
HST_12
Top 200 corpus recall (hp)
HST_13
Top 500 corpus recall (hp)
HST_14
Top 1000 corpus recall (hp)
HST_15
Top 100 queries precision (hp)
HST_16
Top 200 queries precision (hp)
HST_17
Top 500 queries precision (hp)
HST_18
Top 1000 queries precision (hp)
HST_19
Top 100 queries recall (hp)
HST_20
Top 200 queries recall (hp)
HST_21
Top 500 queries recall (hp)
HST_22
Top 1000 queries recall (hp)
HST_23
Entropy (hp)
HST_24
Independent LH (hp)
HMG_25
Number of words in the page (page with max PageRank in the host = mp)
HMG_26

Number of words in the title (mp)
HMG_27
Average word length (mp)
HMG_28
Fraction of anchor text (mp)
HMG_29
Fraction of visible text (mp)
HMG_30
Compression rate (mp)
HMG_31
Top 100 corpus precision (mp)
HMG_32
Top 200 corpus precision (mp)
HMG_33
Top 500 corpus precision (mp)
HMG_34
Top 1000 corpus precision (mp)
HMG_35
Top 100 corpus recall (mp)
HMG_36
Top 200 corpus recall (mp)
HMG_37
Top 500 corpus recall (mp)
HMG_38
Top 1000 corpus recall (mp)
HMG_39
Top 100 queries precision (mp)
HMG_40
Top 200 queries precision (mp)
HMG_41
Top 500 queries precision (mp)
HMG_42
Top 1000 queries precision (mp)
HMG_43
Top 100 queries recall (mp)
HMG_44
Top 200 queries recall (mp)
HMG_45
Top 500 queries recall (mp)
HMG_46
Top 1000 queries recall (mp)
HMG_47
Entropy (mp)
HMG_48
Independent LH (mp)
AVG_49
Number of words in the page (average value for all pages in the host)
AVG_50
Number of words in the title (average value for all pages in the host)

AVG_51
Average word length (average value for all pages in the host)
AVG_52
Fraction of anchor text (average value for all pages in the host)
AVG_53
Fraction of visible text (average value for all pages in the host)
AVG_54
Compression rate (average value for all pages in the host)
AVG_55

Top 100 corpus precision (average value for all pages in the host)
AVG_56
Top 200 corpus precision (average value for all pages in the host)
AVG_57
Top 500 corpus precision (average value for all pages in the host)
AVG_58
Top 1000 corpus precision (average value for all pages in the host)
AVG_59
Top 100 corpus recall (average value for all pages in the host)
AVG_60
Top 200 corpus recall (average value for all pages in the host)
AVG_61
Top 500 corpus recall (average value for all pages in the host)
AVG_62
Top 1000 corpus recall (average value for all pages in the host)
AVG_63
Top 100 queries precision (average value for all pages in the host)
AVG_64
Top 200 queries precision (average value for all pages in the host)
AVG_65
Top 500 queries precision (average value for all pages in the host)
AVG_66
Top 1000 queries precision (average value for all pages in the host)
AVG_67
Top 100 queries recall (average value for all pages in the host)
AVG_68
Top 200 queries recall (average value for all pages in the host)
AVG_69
Top 500 queries recall (average value for all pages in the host)
AVG_70
Top 1000 queries recall (average value for all pages in the host)
AVG_71
Entropy (average value for all pages in the host)
AVG_72
Independent LH (average value for all pages in the host)
STD_73
Number of words in the page (Standard deviation for all pages in the host)
STD_74
Number of words in the title (Standard deviation for all pages in the host)
STD_75
Average word length (Standard deviation for all pages in the host)
STD_76
Fraction of anchor text (Standard deviation for all pages in the host)
STD_77
Fraction of visible text (Standard deviation for all pages in the host)
STD_78
Compression rate in the home page (Standard deviation for all pages in the host)
STD_79
Top 100 corpus precision (Standard deviation for all pages in the host)
STD_80
Top 200 corpus precision (Standard deviation for all pages in the host)
STD_81
Top 500 corpus precision (Standard deviation for all pages in the host)
STD_82
Top 1000 corpus precision (Standard deviation for all pages in the host)
STD_83
Top 100 corpus recall (Standard deviation for all pages in the host)
STD_84
Top 200 corpus recall (Standard deviation for all pages in the host)
STD_85
Top 500 corpus recall (Standard deviation for all pages in the host)
STD_86
Top 1000 corpus recall (Standard deviation for all pages in the host)
STD_87
Top 100 queries precision (Standard deviation for all pages in the host)
STD_88
Top 200 queries precision (Standard deviation for all pages in the host)
STD_89
Top 500 queries precision (Standard deviation for all pages in the host)
STD_90
Top 1000 queries precision (Standard deviation for all pages in the host)
STD_91
Top 100 queries recall (Standard deviation for all pages in the host)
STD_92
Top 200 queries recall (Standard deviation for all pages in the host)
STD_93
Top 500 queries recall (Standard deviation for all pages in the host)
STD_94
Top 1000 queries recall (Standard deviation for all pages in the host)
STD_95
Entropy (Standard deviation for all pages in the host)
STD_96
Independent LH (Standard deviation for all pages in the host)

Appendix II

Tables

Table1: The obtained results from the proposed method.

Combination (Classifier 1, classifier 2, classifier 3)	FP rate (%)	Precision (%)	Accuracy (%)
NN, Random tree, c4.5	0.74	67.1	95.3
NN, AIRS2,Random forest	0.5	76	95.7
Perceptron,C4.5, AIRS2	0.4	68.3	95
NN, Random forest, c4.5	0.49	75.3	95.6
Bayes net, Random tree, NN	2.4	43.5	94.1

Table2: The obtained results from ensemble classifiers.

Combination	Method	FP rate (%)	Precision (%)	Accuracy (%)
NN, Random tree, c4.5	Vote(Combination Rule= Max Probability)	5.9	29.2	91.3
	Vote(Combination Rule=Min Probability)	3.9	32	93.1
	Stacking(meta-classifier = NN)	2	39.2	93.9
	stacking(meta-classifier = Random tree)	2.1	39.4	93.9
	Grading(meta-classifier = NN)	1.2	58.4	95.1
	Grading(meta-classifier = C4.5)	1.6	52.8	94.8
NN, AIRS2,Random forest	Vote(Combination Rule=Max Probability)	3.5	20.4	92.1
	Vote(Combination Rule=Avg Probability)	1.1	27.8	94.3
	Stacking(meta-classifier = NN)	0.8	65.4	95.2
	stacking(meta-classifier = AIRS2)	33.7	7.7	65.4
	Grading(meta-classifier = NN)	0.7	69.9	95.5
Perceptron,C4.5, AIRS2	Vote(Combination Rule=Max Probability)	8.1	19.9	88.9
	Vote(Combination Rule=Avg Probability)	0.3	58.6	94.7
	Stacking(meta-classifier =Perceptron)	0.1	40	94.8
	Stacking(meta-classifier = AIRS2)	4.7	16.1	91
	Grading(meta-classifier = Perceptron)	1.8	16.9	93.3
	Grading(meta-classifier = c4.5)	1	47	94.5
NN, Random forest, c4.5	Vote(Combination Rule=Avg Probability)	3.2	39.8	93.6
	Vote(Combination Rule=Min Probability)	5.9	29.2	91.3
	Stacking(meta-classifier = c4.5)	2.1	42.7	94.1
	Stacking(meta-classifier = Random forest)	1.5	52.6	94.8
	Grading(meta-classifier = NN)	1	62.9	95.2
	Grading(meta-classifier = c4.5)	1	62.4	95.2
Bayes net, Random tree, NN	Vote(Combination Rule=Max Probability)	4.8	29	91.9
	Vote(Combination Rule=Min Probability)	5.5	26.8	91.3
	Stacking(meta-classifier = NN)	4	25	92
	Stacking(meta-classifier = Random tree)	4.7	20.6	91.4
	Grading(meta-classifier = Bayes net)	5.2	29.9	91.8
	Grading(meta-classifier = NN)	2.4	42.6	93.9

Table3: The obtained results from base classifiers.

Classifier	FP Rate (%)	Precision (%)	Accuracy (%)
NN	3	37.1	93.4
C4.5	1.9	47.7	94.4
Random tree	4.6	25.9	91.7
Random forest	0.9	62	95.2
AIRS2	1.6	34.8	15.4
Perceptron	3.8	13.9	91.6
Bayes network	18.9	15.9	80.1