ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2013.0091

WEB LINK SPAM IDENTIFICATION INSPIRED BY ARTIFICIAL IMMUNE SYSTEM AND THE IMPACT OF TPP-FCA FEATURE SELECTION ON SPAM CLASSIFICATION

S. K. Jayanthi[1] and S. Sasikala[2]

[1]*Department of Computer Science, Vellalar College for Women, India*
E-mail: jayanthiskp@gmail.com

[2]*Department of Computer Science, K.S.R College of Arts and Science, India*
E-mail: sasi_sss123@rediff.com

Abstract

Search engines are the doorsteps for retrieving required information from the web. Web spam is a bad method for improving the ranking and visibility of the web pages in search engine results. This paper addresses the problem of the link spam classification through the features of the web sites. Link related features retrieved from the website are used to discriminate the spam and non-spam sites. AIS inspired algorithms are applied for the dataset and results are evaluated. Artificial immune systems are machine learning systems inspired by the principles of the natural immunology. It comprises of supervised learning schemes which can be adapted for the wide range of the classification problems.UK- WEBSPAM-2007 Dataset [8] is used for the experiments. WEKA [9] is used to simulate the classifiers. Artificial Immune Recognition algorithm seems to perform well than the other classes. Best classification accuracy attained is 98.89 by AIRS1 Algorithm. This seems to be good when comparing with the other classifiers accuracy available on the existing literature.

Keywords:

Web Spam, Search Engine, TPP, FCA, AIRS

1. INTRODUCTION

World Wide Web is a huge, dynamic and complex networked information space. Search engines acts as the doorsteps for many users. It is a program which retrieves the required information based on the query. Results with higher relevancy in terms of content and links will be listed in prioritized manner. Higher relevancy yields top positions and visibility in search engine results page (SERP). Some websites manipulate their contents by applying illegal techniques to boost up their rank and visibility in SERP. This creates higher than the deserved ranking for a website. Manipulating the links of a website would yield higher rank in link based ranking search algorithms such as PageRank and HITS. For classifying the spam and non-spam websites with their link related attributes many classifiers were applied. This paper introduces AIS based classifiers for the web spam detection. Results were good when compared to other conventional classifier such as naive bayes, SVM, J48 available in literature.

2. WORKING SCENARIO

Link spam is defined as links between pages that are present for reasons other than merit. Fig.1 shows one such web link spam website. The site contains stuffed links which lead the user again and again to the same page. Link spam takes advantage of link-based ranking algorithms, which gives websites higher rankings the more other highly ranked websites link to it. These techniques also aim at influencing other link-based ranking techniques such as the HITS algorithm.

Fig.1. Sample Website with Link Spam

Web spam detection through extracting the features from website is done with the help of the machine learning techniques. Many techniques were applied to the extracted features in the existing literature. This paper proposes the artificial immune system based machine learning techniques for web spam classification. Results when compared with other machine learning methods existing in the literature seem to be good. The method of application is illustrated in Fig.2.

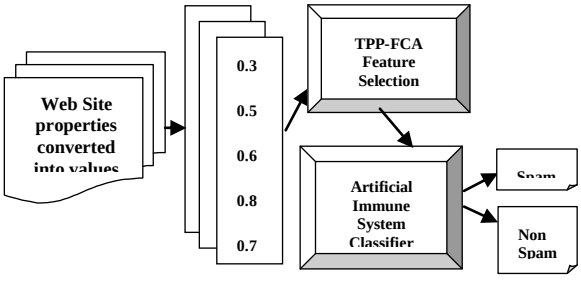


Fig.2. Proposed System

3. RELATED WORK

Shengen et al. [7] propose method for web spam detection, using genetic programming, from existing link-based features and use them as the inputs to support vector machine and genetic programming classifiers. According to the authors, the classifiers

that use the new features achieve better results compared with the features provided in the original database.

Erdelyi et al. [5] used ensemble based methods Bagged LogitBoost, J48 Decision Trees, Bagged Cost-sensitive Decision Trees, Logistic Regression, Random Forests and Naïve Bayes for web spam detection. They conclude that with appropriate learning techniques, a small and computationally inexpensive feature subset outperforms all previous results published so far on their data set and can only slightly be further improved by computationally expensive features. They test their method on two major publicly available data sets, the Web Spam Challenge 2008 data set WEBSPAM-UK2007 and the ECML/PKDD Discovery Challenge data set DC2010.

Kariampor et al. [4] performs classification of web spam using imperialist competitive algorithm and genetic algorithm. Imperialist competitive algorithm is a novel optimization algorithm that is inspired by socio-political process of imperialism in the real world. Experiments are carried out on WEBSPAM-UK2007 data set, which show feature selection improves classification accuracy, and imperialist competitive algorithm outperforms GA.

Geng et al. [6] used re-extracted features based on the host level link graph and the predicted spamicity, clustering, propagation and neighbor details and used WEBSPAM-UK 2006 dataset as a base. They use bagging, a famous meta-learning algorithm with c4.5.

4. PROBLEM DESCRIPTION

Spamdexing subvert the search engine results through manipulating the content, link or meta tags of a website. Content spamdexing is achieved through the interpretation of the title text, anchor text or body text of a webpage. One example is stuffing a popular keyword in any part of webpage. Link spamdexing refers manipulation of the links (inlinks and outlinks). Thus spamdexing of a website *W* is referred as:

$$spam(W) = \forall_{WP \in W} \left(\begin{array}{l} \sum_{i=1}^{N} CS'(W) + \sum_{i=1}^{N} LS'(W) + \\ \sum_{i=1}^{N} MS'(W) \end{array} \right) \quad (1)$$

where, WP – webpages in a particular website *W*, *n* – number of pages, CS' – content spammed, LS' – link spammed, MS' –meta spammed. With the help of computed link based features the classification is performed.

5. DATA ENGINEERING

5.1 OVERVIEW OF THE UK-WEBSPAM 2007 DATASET

UK-WEBSPAM-2007 dataset [8] is based on a set of pages obtained from a crawler of the .uk domain. The set includes 77.9 million pages, corresponding to 11402 hosts, among which over 8000 hosts have been labelled as “spam”, “nonspam” or “borderline”. The link based feature set contains originally 3998 instances with 44 attributes.

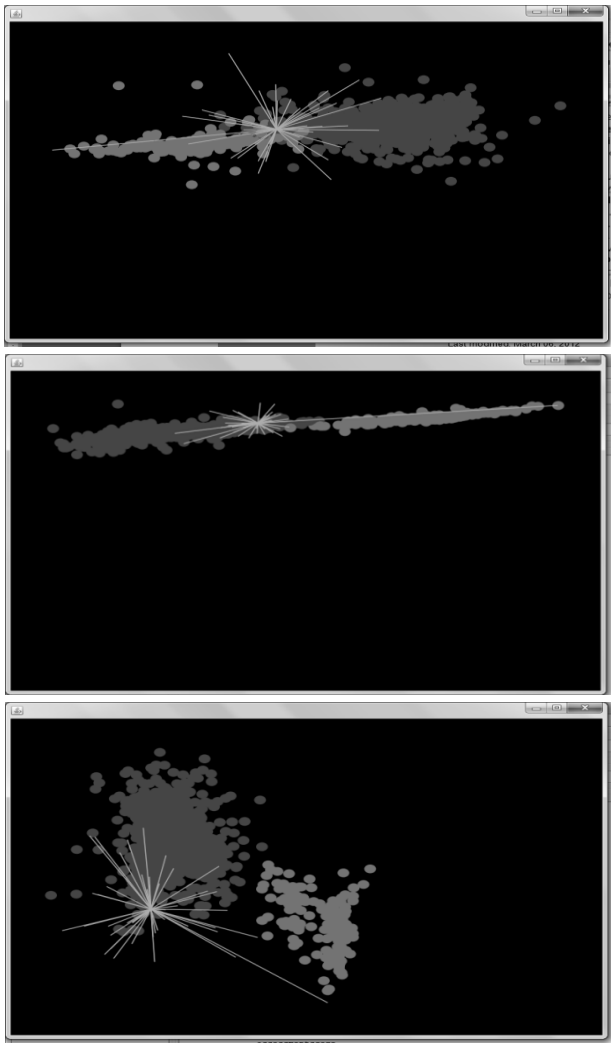
Table.1. WEBSPAM-UK-2007 and 2006 dataset comparison

Year	2006	2007
Number of nodes(Hosts)	11,402	114,529
Number of Edges	730,774	1,836,441
Number of labelled Host	10,662	8,479

5.2 FEATURE SELECTION WITH TPP-FCA

5.2.1 Targeted Projection Pursuit – TPP:

Targeted projection pursuit (TPP) is a type of statistical technique used for exploratory data analysis, information visualization, and feature selection. It allows the user to interactively explore very complex data to find features or patterns of potential interest. Conventional, or 'blind', projection pursuit, finds the most "interesting" possible projections in multidimensional data, using a search algorithm that optimizes some fixed criterion of "interestingness" – such as deviation from a normal distribution.



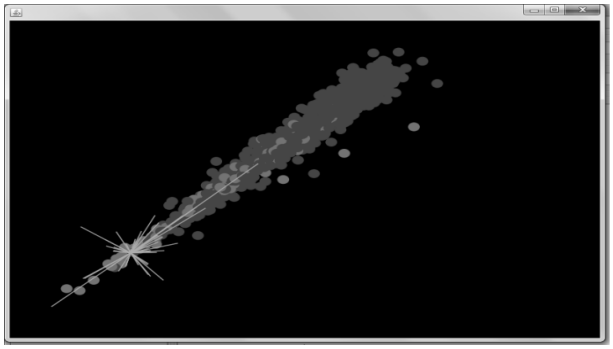


Fig.3. Four different Perspective of the dataset in TPP

In contrast, targeted projection pursuit allows the user to explore the space of projections by manipulating data points directly in an interactive scatter plot [10]. The UK-WEBSPAM-2007 link based features is subject to the standard 10-fold cross validation of the TPP. Weka is used to perform the feature selection. The highly influential features are selected and used further in the experiments. Four different perspectives of the feature selection on the base dataset is given in Fig.3. Original dataset contains of 44 attributes and after subject to TPP 10 attributes were selected. Results of the TPP feature selection is given below:

TPPAttributeSearch

Attribute Evaluator (supervised, Class (nominal): 43 class):

weka.attributeSelection.TPPAttributeEvaluation@b7b80

Ranked attributes:

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

siteneighbors_2_hp

assortativity_hp

pagerank_hp

trustrank_hp

outdegree_hp

reciprocity_hp

avgin_out_hp

indegree_hp

siteneighbors_1_hp

5.2.2 Formal Concept Analysis – FCA:

Formal concept analysis is a principled way of deriving a concept hierarchy or formal ontology from a collection of objects and their properties. Each concept in the hierarchy represents the set of objects sharing the same values for a certain set of properties; and each sub-concept in the hierarchy contains a subset of the objects in the concepts above it. The aim and meaning of Formal Concept Analysis as mathematical theory of concepts and concept hierarchies is to support the rational communication of humans by mathematically developing appropriate conceptual structures which can be logically activated [10].

5.2.3 Contexts and concepts:

A (formal) context consists of a set of objects O, a set of unary attributes A, and an indication of which objects have which attributes. Formally it can be regarded as a bipartite graph $I \subseteq O \times A$.

A (formal) concept for a context is defined to be a pair (O_i, A_i) such that,

1. $O_i \subseteq O$ (objects of the dataset)
2. $A_i \subseteq A$ (attributes of the dataset)
3. every object in O_i has every attribute in A_i
4. for every object in O that is not in O_i, there is an attribute in A_i that the object does not have
5. for every attribute in A that is not in A_i, there is an object in O_i that does not have that attribute

O_i is called the extent of the concept, A_i the intent.

A context may be described as a table, with the objects corresponding to the rows of the table, the attributes corresponding to the columns of the table, and a Boolean value (in the experiment represented graphically as a checkmark) in cell (x, y) whenever object x has value y. Generated context of the spam classification is given in Fig.4.

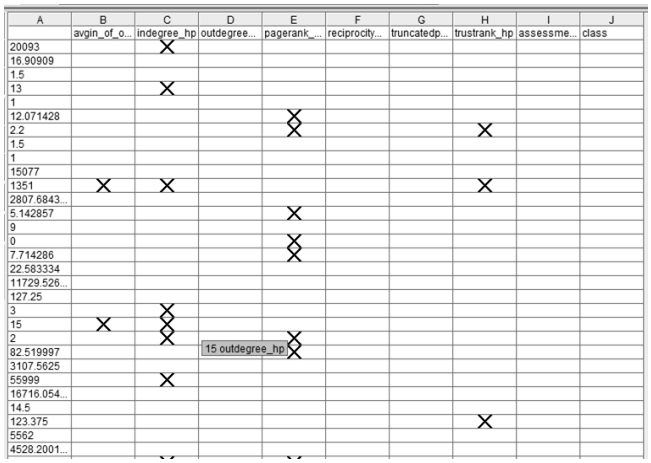


Fig.4. Context of the $TPP_{Dataset}$ after FCA

5.2.4 Concept and Concept Lattice:

A concept, in this representation, forms a maximal sub array such that all cells within the sub array are checked. The concepts (O_i, A_i) defined above can be partially ordered by inclusion: if (O_i, A_i) and (O_j, A_j) are concepts, we define a partial order $\leq$ by saying that $(O_i, A_i) \leq (O_j, A_j)$ whenever $O_i \subseteq O_j$. Equivalently, $(O_i, A_i) \leq (O_j, A_j)$ whenever $A_j \subseteq A_i$. Every pair of concepts in this partial order has a unique greatest lower bound (meet). The greatest lower bound of (O_i, A_i) and (O_j, A_j) is the concept with objects $O_i \cap O_j$; it has as its attributes the union of A_i, A_j, and any additional attributes held by all objects in $O_i \cap O_j$. Symmetrically, every pair of concepts in this partial order has a unique least upper bound (join). The least upper bound of (O_i, A_i) and (O_j, A_j) is the concept with attributes $A_i \cap A_j$; it has as its objects the union of O_i, O_j, and any additional objects that have all attributes in $A_i \cap A_j$. These meet and join operations satisfy the axioms defining a lattice. Any finite lattice may be generated as the concept lattice for some context. The concept lattice which is created for spamdexing features is given in Fig.5. For, let L be a finite lattice, and form a context in which the objects and the attributes both correspond to elements of L. In this

context, let object x have attribute y exactly when x and y are ordered as $x \leq y$ in the lattice [11].

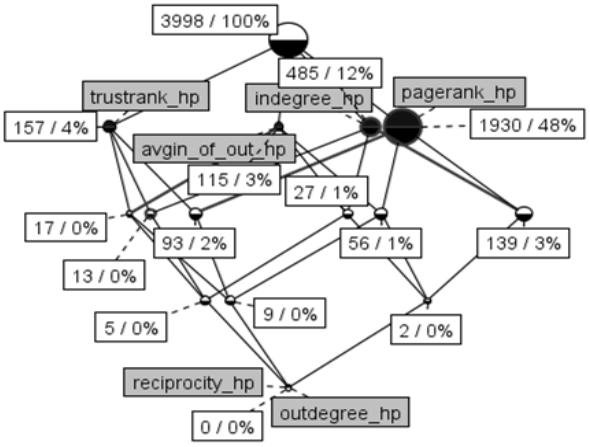


Fig.5. Concept Lattice obtained $TPP_{Dataset}$ after FCA

5.2.5 Formal Concept Analysis - Selected Features:

Features selected after FCA is given below:

avgin_of_out_hp

indegree_hp

outdegree_hp

pagerank_hp

reciprocity_hp

trustrank_hp

class

assessmentscore

Algorithm applied for the feature selection is given and balanced and unbalanced dataset were used for the experiments.

Algorithm 1: TPP-FCA Feature Selection

Description:

Original WEBSPAM-UK-2007 link based features dataset contains 44 attributes and it is unbalanced. In order to find the most effective features from the dataset TPP and FCA feature selection methods are applied and new sets of data are formed.

Step 1: Apply the Targeted Projection Pursuit with standard 10-fold cross validation.

Step 2: Select attributes with good influence over spamdexing classification. The resultant dataset obtained is named as U-$TPP_{Dataset}$ (SET 1). The dataset is unbalanced.

Step 3: Perform BCC and create balanced dataset: B-$TPP_{Dataset}$ (SET 2)

Step 3: Apply Formal Concept Analysis to $TPP_{Dataset}$ to obtain the highly effective features from the selected attributes set.

Step 4: Concepts were built and highly effective features correlation is visualized. The resultant dataset from step 3 is named as U-TPP + $FCA_{Dataset}$ (SET 3). This dataset is unbalanced.

Step 5: Perform BCC and create balanced dataset: B-TPP + $FCA_{Dataset}$ (SET 4)

Step 6: Subject the result of step 2, 3, 4 and 5 to the AIRS classifier and obtain the result.

Algorithm 2: BCC- Balanced Containers Creation

Description:

The number of instances present in the original $TPP_{Dataset}$ and TPP + $FCA_{Dataset}$ are unbalanced. The number of samples representing the non-spam are 70% and spam are 30%. Start creating balanced containers with samples of both kinds equally 50% non-spam and 50% spam by the following steps.

Step 1: Categorize the spam and non-spam samples separately.

Step 2: Arrange the spam samples in high-to-low assessment score order.

Step 3: Arrange the non-spam samples in low-to-high assessment score order.

Step 4: Place first 200 instances of the spam samples with first 200 instances of the non-spam samples and create two containers with 400 instances each.

6. ARTIFICIAL IMMUNE SYSTEM AND PROPOSED CLASSIFIERS

Artificial Immune Systems (AIS) are adaptive machine learning systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving. In this paper, six algorithms were evaluated in three categories of the AIS.

Category 1: Artificial Immune Recognition Algorithms – AIRS1, AIRS2Parallel [1]

Category 2: Clonal Selection Algorithms – CLONALG, CSCA[2]

Category 3: Immunity based Algorithms – Immunos1, Immunos99 [3]

The first category is Artificial Immune Recognition systems. The main task of the immune system of an organ is to detect the pathogens (harmful material) and combat against that in order to protect the organ. Antigen is a substance that evokes the production of one or more antibodies. Antigens role is to neutralize the effect of the pathogen. The anomaly detection is performed with the help of B-Cells and T-Cells. B cells belong to a group of white blood cells known as lymphocytes, making them a vital part of the immune system. T cells or T lymphocytes belong to a group of white blood cells known as lymphocytes, and play a central role in cell-mediated immunity. The algorithm of the AIRS category is as follows:

Algorithm 1: AIRS

Start

Phase I: Antigen Selection

Start: Training and Best match memory cell (Antigen) selection

1. Normalize the training data by selecting representative antigens through affinity measure.
2. Apply Distance Measure

$$E - Dist = \sqrt{\sum_{i=1}^{N} (v1_i.v2_i)^2} \quad (2)$$

where, $v1$ and $v2$ represent two elements that affinity is measured between and n is the number of attributes.

3. Calculate maximum distance between two data vectors by root of the sum of the square ranges, where r is the known data range for attribute i.

$$maxDist = \sqrt{\sum_{i=1}^{N} (r_i)^2} \quad (3)$$

4. Calculate Affinity measure which is a similarity value

$$Affinity = \frac{E - Dist}{maxDist} \quad (4)$$

5. Select the memory cell with the greatest stimulation which can be used for affinity maturation process.

$$Stim = 1 - Affinity \quad (5)$$

6. Calculate the number of mutated clones created of the best match as follows:

$$numClones = stim.clonalRate.hyperMutationRate \quad (6)$$

7. Mutated clones of best match memory cell are refined and added to ARB (Artificial Recognition Ball) pool.
8. Refinement completed and memory cell candidate selected.

Stop: Training process completed

Phase II: Classification

Start: Classification based on selected best match memory cell begins

1. K-nearest neighbor approach used
2. Selected memory cells matched with the rest of the dataset
3. Instances are classified

Stop: Classification summary listed

End

The two variants used in this category are AIRS1 and AIRS2Parallel the specification of the parameters are explained in section 6.In AIRS2Parallel, instead of being distributed across multiple processes, this implementation allows AIRS to be executed by multiple threads.

The second category deals with the clonal selection criteria. Clonal selection theory immunity can be acquired using B-cells and T-cells in response to the antigens over time called affinity maturation. Darwinian theory is applied here where selection is carried out by affinity-antibody interactions, reproduction through cell division and variation through somatic hypermutation. The algorithm for the clonal selection category is as follows:

Algorithm 2: Clonal Selection based

Start:

1. Create a pool of antibodies, N
2. For G generations
3. For all antigens
   a. Select an antibody in random
   b. Select number of clones created from each of the n selected antibodies as follows:

$$numClones = \left\lfloor \frac{\beta.N}{i} + 0.5 \right\rfloor \quad (7)$$

where, β is the clonal factor, N is the size of the antibody pool, and i is the antibody current rank where $i \in [1, n]$.

   c. Calculate the total number of clones prepared for each antigen exposure to the system as:

$$N_c = \sum_{i=1}^{N} \left\lfloor \frac{\beta.N}{i} + 0.5 \right\rfloor \quad (8)$$

where, N_c is the total number of clones, and n is the number of selected antibodies.

   d. Calculate Affinity for antigen as said in the Algorithm 1
   e. Select n antibodies with best affinity
   f. Generate clones of the selected antibodies and mutate
   g. Calculate affinity for entire clonal set
   h. Select best match memory cell
   i. Compare it with the rest of the dataset
4. Classification results were given

End

Variants of the clonal selection algorithm used in this work are CLONALG and CSCA. CLONALG (CLONal selection ALGorithm) is based on the clonal selection theory of acquired immunity. Clonal Selection Classification Algorithm (CSCA) represents algorithm based on abstractions of the clonal selection theory of acquired immunity and the CLONALG technique.

The third category of the algorithm is based on immunity structure identification. Antigens are able to improve themselves adapting to provide an increasingly stronger and rapid response. The two main cells involved in this process are B-cells and T-cells. When a T-cell or a B-cell encounters an antigen, and has a sufficient affinity with its surface receptors, the cell becomes activated. The cell binds to the antigen though this step alone is not sufficient to elicit an immune response [2]. The algorithm for this category is as follows:

Algorithm 3: Immunity based

Start:

1. Collect the available antigen types
2. Categorize them and process each of them
3. Create a T-cell to represent the group
4. Match and pick appropriate antigens
5. For each antigen group
   a. Create a B-cell to represent the subgroup.
   b. Calculate affinity between a single B-cell and an unknown antigen as follows:

$$affinity = \sqrt{\sum_{A}^{i=1} af_i} \quad (10)$$

where, A is the total number of attributes in the data vectors and af_i is the affinity of the i^{th} attribute.

   c. Match and pick appropriate B-cell antigens and consolidate the selected clones
6. Classification results given

Stop

Variants of the immunity based algorithms used in this work are: Immunos1 and Immunoos99. Immunos1 assumes no data reduction, thus the clone population prepared is maintained and is used to classify unknown data instances. This naive approach is provided as a baseline for performance, and is very similar to the k-nearest neighbor algorithm. Immunos99 has integrated cell-proliferation and hyper mutation techniques from other immune-inspired classification systems. It also has superior data-reduction capabilities [3].

7. EXPERIMENTAL RESULTS

As stated earlier six AIS based algorithms were implied for the UK-WEBSPAM-2007 dataset. Results were promising. Among the six methods AIRS1 and AIRS2Parallel performs well. They offer maximum accuracy for classification. The settings used by six algorithms and results were listed below.

7.1 AIRS1

AIRS1 algorithm is evaluated with affinity threshold value 0.2, initial pool size is set to 1 and clonal rate is set to 10.0, Hypermutationrate is 2.0, Knn is 3, Mutationrate is 0.1 and Stimulation value is set to 0.9. The training data summary is as follows:

–Training Summary –	
Affinity Threshold:	*0.217*
Total training instances:	*3,998*
Total memory cell replacements:	*447*
Mean ARB clones per refinement iteration:	*124.387*
Mean total resources per refinement iteration:	*147.826*
Mean pool size per refinement iteration:	*141.352*
Mean memory cell clones per antigen:	*18.291*
Mean ARB refinement iterations per antigen:	*1.376*
Mean ARB prunings per refinement iteration:	*138.377*
–Classifier Statistics–	
Data Reduction Percentage:	*91.046%*
–Classifier Memory Cells–	
Total:	*358*
nonspam:	*307*
spam:	*51*
Time taken to build model:	*13.98 sec*

The ROC curve of the classifier is given in Fig.6.

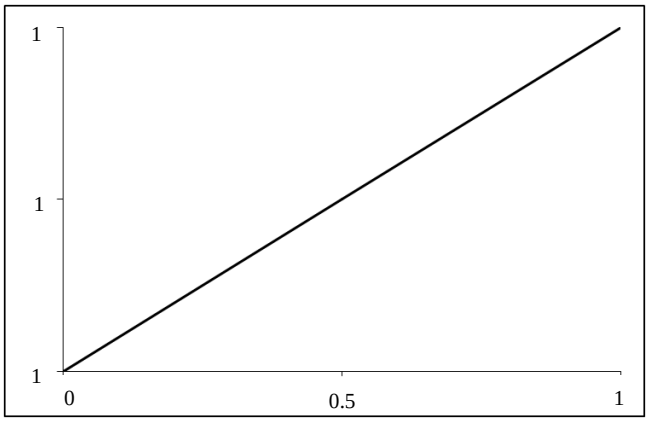


Fig.6. Plot: Area under ROC 0.903 for AIRS1

7.2 AIRS2PARALLEL

AIRS2Parallel is evaluated with affinity threshold scalar value 0.2, Clonalrate is 10.0, Hypermutationrate is 2.0, Knn is 3, Meminitialpoolsize is 1, Mergemode adopted is concatenate and prune, Numinstanceaffinity threshold is -1, Numthreads 2, Seed 1 and Stimulation value 0.9. The training data summary after pruning is as follows:

–Training Summary –	
Affinity Threshold:	*0.217*
Total training instances:	*1,999*
Total memory cell replacements:	*1,595*
Mean ARB clones per refinement iteration:	*48.57*
Mean total resources per refinement iteration:	*123.317*
Mean pool size per refinement iteration:	*66.122*
Mean memory cell clones per antigen:	*18.078*
Mean ARB refinement iterations per antigen:	*2.005*
Mean ARB prunings per refinement iteration:	*50.102*
–Classifier Statistics–	
Data Reduction Percentage:	*92.396%*
–Classifier Memory Cells–	
Total:	*304*
nonspam:	*281*
spam:	*23*
Time taken to build model:	*27.02 sec*

The ROC curve of the classifier is given in Fig.7.

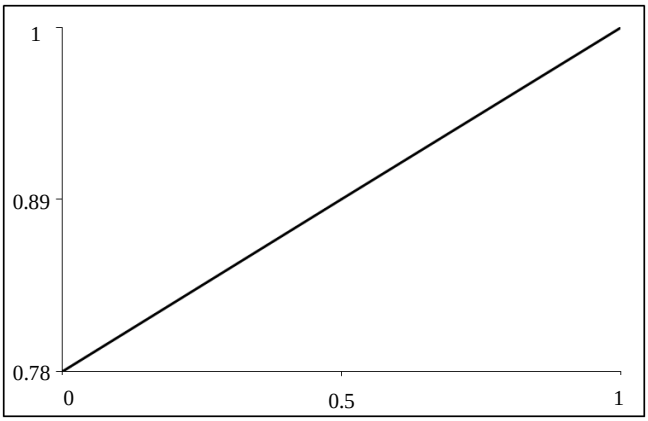


Fig.7. Plot:Area under ROC 0.8919 for AIRS2Parallel

7.3 CLONALG

CLONALG clonal selection algorithm is evaluated with the parameters such as Antibodypool size 30, Clonalfactor 0.1, Numgenerations 10, Seed 1 and Selection pool size 20. The ROC curve of the classifier is given in Fig.8.

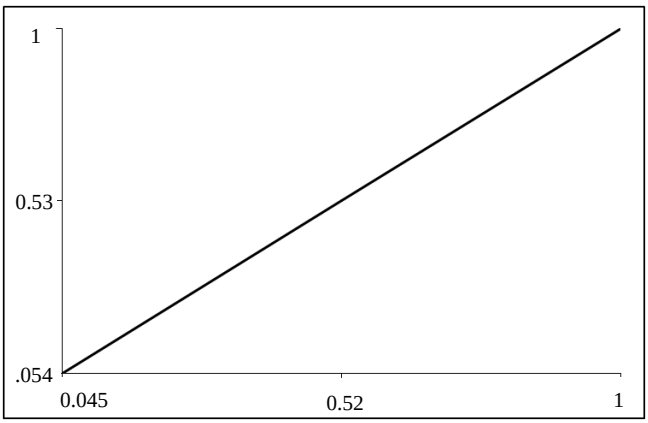


Fig.8. Plot:Area under ROC 0.5045 for CLONALG

7.4 CSCA

CSCA is evaluated with parameters: Knn 1, Clonal scale factor 1.0, Initial population size 50, Minimum fitness threshold 1.0, Numpartitions 1, Seed 1 and Total generations 5. The ROC curve of the classifier is given in Fig.9. The training summary is as follows:

–Training Summary –

Generations Completed:	*5*
Antibodies pruned per generation:	*3,140.2 (1,573.931)*
Antibodies without error per generation:	*357.6 (198.556)*
Population size per generation:	*4,537.2 (252.324)*
Antibody fitness per generation:	*5.184 (8.868)*
Antibody class switches per generation:	*25 (15.786)*
Selection set size per generation:	*89.8 (28.28)*
Training accuracy per generation:	*93.732 (0.516)*
Inserted antibodies per generation:	*89.8 (28.28)*
Cloned antibodies per generation:	*4,000 (3.847)*
–Classifier Summary–	
Data Reduction Percentage:	*81.791%*
Total Training Instances:	*3998*
Total antibodies:	*728*
–Classifier Memory Cells–	
nonspam:	*727*
spam:	*1*
Time taken to build model:	*135.19 sec*

The ROC curve of the classifier is given in Fig.9.

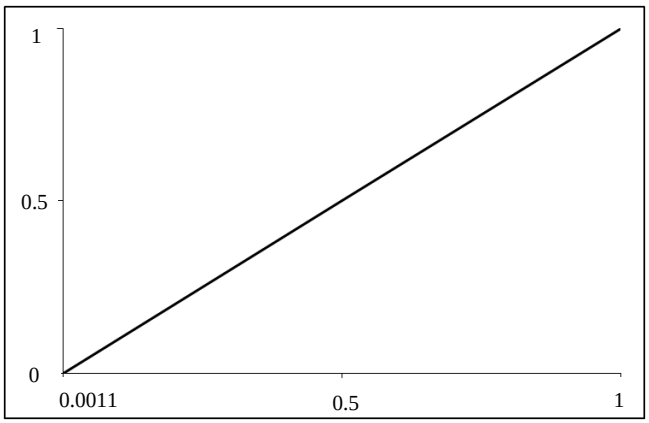


Fig.9. Plot: Area under ROC 0.4995 for CSCA

7.5 IMMUNOS99

Immunos99 is evaluated with the parameters Minimum fitness threshold -1, Seed 1, Seed population percentage 0.2 and Total generations 1. The training summary is as follows:

–Training Summary –

Group Name:	*nonspam*
Cells pruned per generation:	*0 (0)*
Population size per generation:	*4,529 (0)*
Cell fitness per generation:	*17.145 (0)*
Cloned cells per generation:	*3,776 (0)*
Cells deleted in final prune:	*3,734*
Group Name:	*spam*
Cells pruned per generation:	*25 (0)*
Population size per generation:	*270 (0)*
Cell fitness per generation:	*0.059 (0)*
Cloned cells per generation:	*223 (0)*
Cells deleted in final prune:	*223*
–Classifier Summary–	
Data Reduction Percentage:	*78.939%*
Total Training Instances:	*3998*

Total cells: *842*

–Classifier Memory Cells–

nonspam: *795*

spam: *47*

Time taken to build model: *59.1 sec*

The ROC curve of the classifier is given in Fig.10.

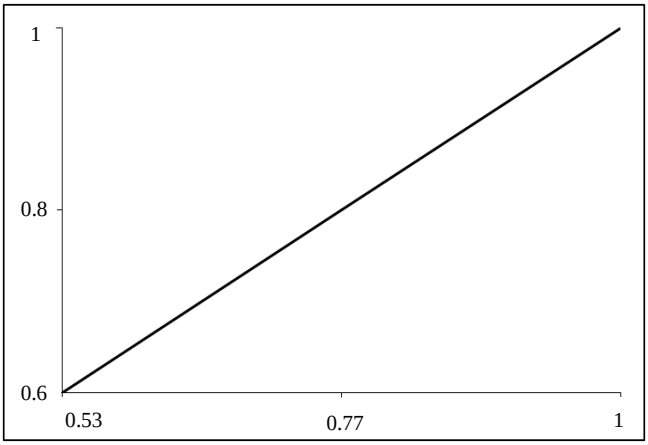


Fig.10. Plot: Area under ROC 0.5334 for Immunos99

7.6 IMMUNOS1

The Immunos1 is evaluated for the given dataset and generated ROC is given below.

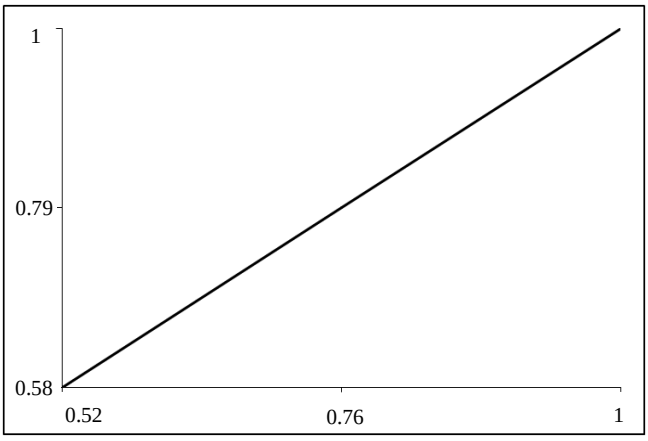


Fig.11. Plot: Area under ROC 0.5285 for Immunos1

8. EVALUATION METRICS

The area under curve of the ROC will be a better evaluation metric to predict the classifier performance. The ROC curves generated by the six AIS classifiers are given in Fig.3 to Fig.7. Based on the AUC values the AIRS1 and AIRS2Parallel classifiers perform well. Every classifier will generate a confusion matrix which gives the misnomers in the predictions. The classifiers used in the work generate the confusion matrix and the specification and formulas are given in Table.1. The generated confusion matrix values for all the six AIS classifiers are listed in Table.2.

Table.2. Confusion Matrix Specification

		Actual outcome			
		P	N		
Test outcome	P	**a**	**b**	PPV	**a/(a+b)**
	N	**c**	**d**	NPV	**d/(c+d)**
		α	β		
		a/(a+c)	**d/(b+d)**		

P-Positive N-Negative
PPV - Positive Predictive Value
NPV - Negative Predictive Value
α – Sensitivity β – Specificity

Table.3. Experimental results of six AIS Algorithms

Confusion Matrices					
AIRS1		**Actual**			
		P	N		
Test outcome	P	3775	1	**PPV**	0.9997
	N	43	179	**NPV**	0.8063
		α	B		
		0.9887 37559	0.9944 44444		
AIRS2Parallel		**Actual**			
		P	N		
Test outcome	P	3776	0	**PPV**	1
	N	48	174	**NPV**	0.7837
		α	B		
		0.9874 47699	1		
CLONALG		**Actual**			
		P	N		
Test outcome	P	3606	170	**PPV**	0.9549
	N	210	2	**NPV**	0.0094
		α	B		
		0.9449 68553	0.0116 27907		
CSCA		**Actual**			
		P	N		
Test outcome	P	3772	4	**PPV**	0.9989
	N	222	0	**NPV**	0
		α	B		

		0.9444 16625	0		
Immunos1		**Actual**			
		P	N		
Test outcome	P	1814	1962	**PPV**	0.4804
	N	94	128	**NPV**	0.5765
		α	B		
		0.9507 33753	0.0652 39551		
Immunos99		**Actual**			
		P	N		
Test outcome	P	1766	2010	**PPV**	0.4676
	N	89	133	**NPV**	0.5990
		α	β		
		0.9520 21563	0.0620 62529		

From the table it is visible that AIRS1 and AIRS2Parallel seems to have good sensitivity and specificity. The PPV and NPV prediction is high for AIRS1, AIRS2Parallel and CLONALG. Hence it is clear that the discriminative ability of AIRS1 is up to the mark. The F-measure for the Spam and non spam (ham) is portrayed in Fig.12. For predicting both classes the AIRS1 and AIRS2Parallel classifiers of category 1 performs well than the others.

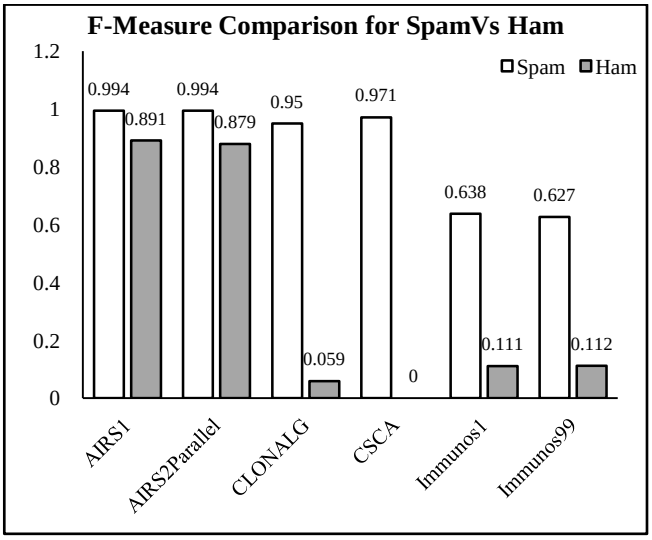


Fig.12. F-Measure comparison for Spam and Ham

The Fig.13 depicts the True Positive Rate (TPR), False Positive Rate (FPR), Precision and Recall comparison of the considered classifiers. The Fig.14 gives the error rate of the classifiers. Immunos1 has the highest error rate followed by Immunos99 and hence the immunity based cloning couldn't be much effective in web spam classification.

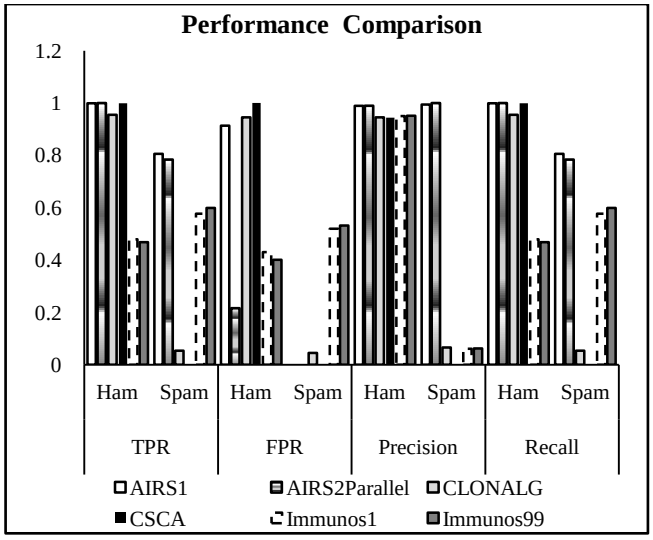


Fig.13. Performance Comparison of AIS methods

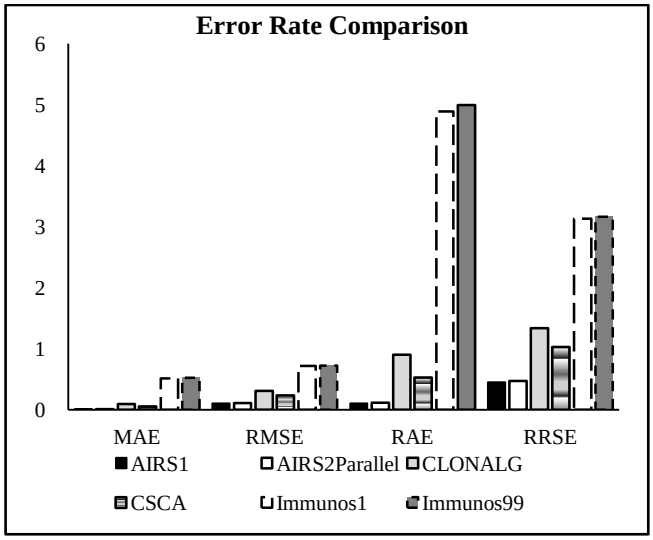


Fig.14. Error rate comparison of the AIS methods

9. RESULTS SUMMARY AND DISCUSSION

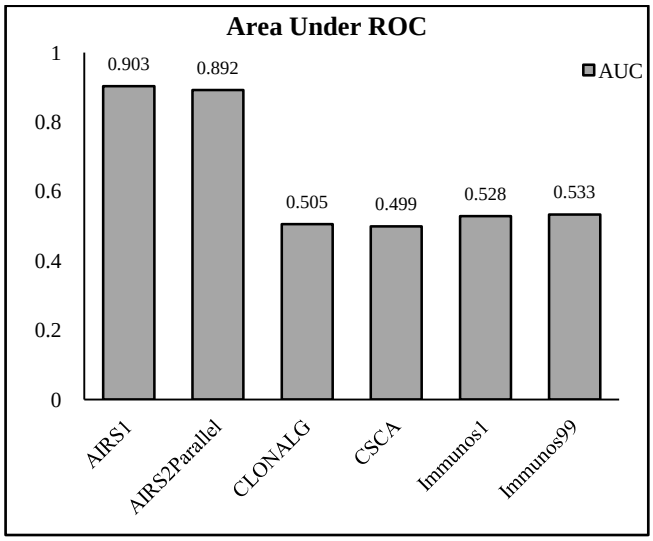


Fig.15. AUC for AIS Methods

The individual ROC curves are given in section 6 and the overall comparison of the six classifiers is depicted in Fig.15. As stated the AIRS1 is leading in the AUC value followed by AIRS2Parallel. Time taken for the classification task is depicted in Fig.16. CSCA algorithm takes maximum time 136 seconds followed by Immunos99 algorithm. In time factor also the AIRS1 and AIRS2Parallell seems to be good.

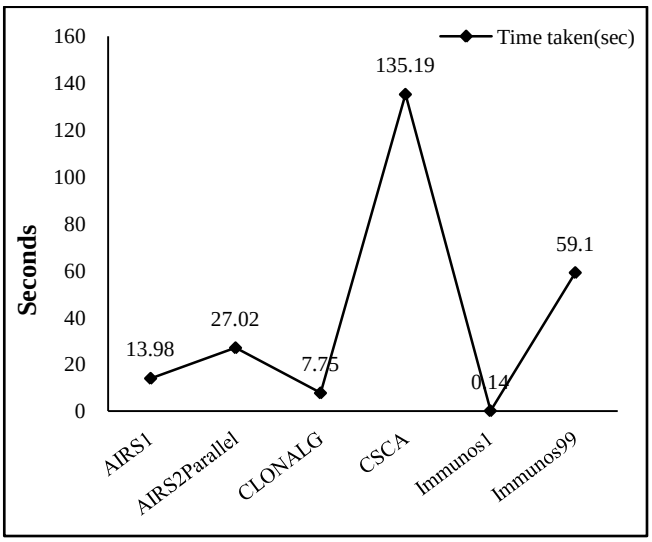


Fig.16. Time Taken for the proposed classifiers

The summary of the TPR, FPR, Precision and Recall for spam and ham (non spam) values are given in Table.3 and Table.4 respectively. Number of correctly classified instances in the given dataset and incorrectly classified instances were tabulated in Table.6. Error rates were listed in Table.7. Comparison of the AIS results with other existing methods in literature has been given in Table.8.

Table.4. True Positive and False Positive Rate for the Spam and Ham in AIS

Method	TPR		FPR	
	Ham	Spam	Ham	Spam
AIRS1	1	0.806	0.914	0
AIRS2Parallel	1	0.784	0.216	0
CLONALG	0.955	0.054	0.946	0.045
CSCA	0.999	0	1	0.001
Immunos1	0.48	0.577	0.43	0.52
Immunos99	0.468	0.599	0.401	0.532

Table.5. Precision, Recall for the Spam and Ham in AIS

Method	Precision		Recall	
	Ham	Spam	Ham	Spam
AIRS1	0.989	0.994	1	0.806
AIRS2Parallel	0.989	1	1	0.784
CLONALG	0.945	0.066	0.955	0.054
CSCA	0.944	0	0.999	0
Immunos1	0.951	0.061	0.48	0.577
Immunos99	0.952	0.062	0.468	0.599

Table.6. F-Measure, AUC for AIS

Method	F-Measure		AUC
	Ham	Spam	
AIRS1	0.994	0.891	0.903
AIRS2Parallel	0.994	0.879	0.892
CLONALG	0.95	0.059	0.505
CSCA	0.971	0	0.499
Immunos1	0.638	0.111	0.528
Immunos99	0.627	0.112	0.533

Table.7. Correctly classified instances (CCI) and Incorrectly classified instances (ICI) in AIS

Method	CCI	ICI	Accuracy (%)
AIRS1	3954	44	98.89
AIRS2Parallel	3950	48	98.29
CLONALG	3618	380	90.49
CSCA	3772	226	94.34
Immunos1	1942	2056	48.57
Immunos99	1899	2099	47.49

Table.8. Error rate in AIS (Mean Absolute Error-MAE, Root Mean Squared Error-RMSE, Root Absolute Error-RAE, Root Relative Squared Error-RRSE)

Method	MAE	RMSE	RAE	RRSE
AIRS1	0.011	0.1049	0.104	0.45
AIRS2Parallel	0.012	0.109	0.114	0.47
CLONALG	0.095	0.308	0.904	1.34
CSCA	0.056	0.237	0.53	1.03
Immunos1	0.514	0.717	4.89	3.13
Immunos99	0.525	0.724	4.99	3.16

Table.9. Comparison of results with existing literature

Method	Feature set	F-Measure	AUC
Erdelyi et.al	Link		0.759
Shengen et.al	Link	0.726	
Jabber et al	Link	0.882	

Proposed Method - AIRS1	Link	0.9425	0.903
Proposed Method - AIRS2Parallel	Link	0.9365	0.892

Comparison of the AIS results with other existing methods in literature has been given in Table.8. It is clearly visible that the for the link based features the AIS based classifiers yields highest performance when compared with the traditional classifiers such as decision trees, naive bayes, SVM. The projection plot of the spam and nonspam samples based on the used dataset is given in Fig.18. The ROC curves discussed in section 6 depicts the spam occurrences the overall comparison of the six classifiers with both spam and non spam AUC values depicted in single simulated graph is given in Fig.19. The knowledge flow layout used for the above ROC curve generation is as follows in Fig.17.

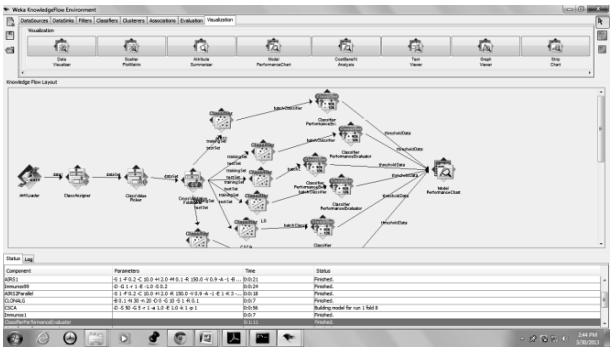


Fig.17. KL Layout for ROC Curves

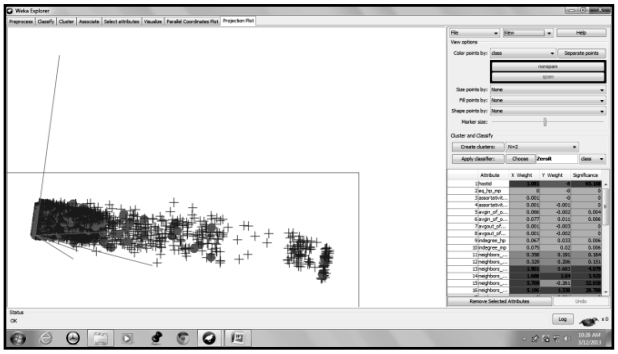


Fig.18. Projection Plot of the data using the Principle components spam vs. non spam

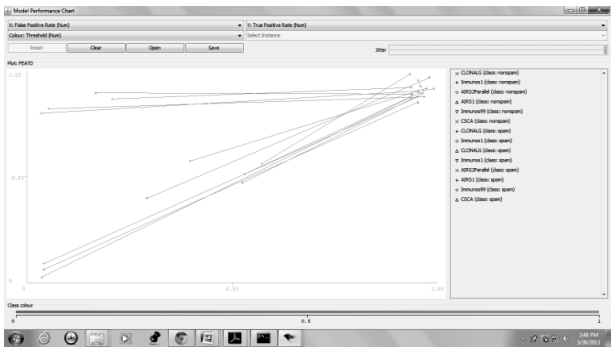


Fig.19. Overall comparison of all classifiers for the spam and non spam

10. CONCLUSION AND FUTURE WORK

Search Engines focus on the value of the time spent by the user before them. Hence when the user got frustrated with the results reliability it may affect the search engines credibility and income. Combating spamdexing is a crucial need of the hour in search engines. This paper addresses the problem of the link spam classification through the features of the web sites. Link related features retrieved from the website can be used to discriminate the spam and non-spam sites. AIS inspired algorithms are applied for the dataset and results are evaluated. Best classification accuracy attained is 98.89 by AIRS1 Algorithm. This seems to be good when comparing with the other classifiers accuracy available in literature. This paper considers the existing dataset and evaluates the classifiers on them. It is planned to collect real time data for a suspicious website and convert the values into a database. Then the database could be used for the website classification. This could be the future enhancement. Combining the content based features with the dataset could give more accuracy. This paper only focus on link based features. Hence content spam cannot be identified. When both content and link based features were used it could be more effective collaborative filter and classifier. That could also be the future scope of the paper.

APPENDIX – A

Sample Dataset

Base Dataset Features

{

hostid, eq_hp_mp assortativity_hp assortativity_mp avgin_of_out_hp avgin_of_out_mp avgout_of_in_hp avgout_of_in_mp indegree_hp indegree_mp neighbors_2_hp neighbors_2_mp neighbors_3_hp neighbors_3_mp neighbors_4_hp neighbors_4_mp outdegree_hp outdegree_mp pagerank_hp pagerank_mp prsigma_hp prsigma_mp reciprocity_hp reciprocity_mp siteneighbors_1_hp siteneighbors_1_mp siteneighbors_2_hp siteneighbors_2_mp siteneighbors_3_hp siteneighbors_3_mp siteneighbors_4_hp siteneighbors_4_mp truncatedpagerank_1_hp truncatedpagerank_1_mp truncatedpagerank_2_hp truncatedpagerank_2_mp truncatedpagerank_3_hp truncatedpagerank_3_mp truncatedpagerank_4_hp truncatedpagerank_4_mp trustrank_hp trustrank_mp class assessmentscore

}

TPP Selected Features

{

assortativity_hp, siteneighbors_2_hp, neighbors_2_hp, avgin_of_out_hp, indegree_hp, outdegree_hp, pagerank_hp, reciprocity_hp, truncatedpagerank_1_mp, trustrank_hp, assessmentscore, class

0.613757,17,69,2.2,24,5,0,1,0,0,1, spam

0.002695,1,2,1351,1,1,0,0,1,spam

2.339399,1,93,123.375,74,15,0,0.125,0,0,1,spam

6.863007,25,10622,33047.5,2941,33,0.000001,0.764706,0.000001 0,0.75,spam

0.42328,1,16,122.800003,16,14,0,0.133333,0,0,1,spam

}

FCA Selected Features

avgin_of_out_hp, indegree_hp, outdegree_hp, pagerank_hp, reciprocity_hp, trustrank_hp, class, assessmentscore

{

2.2, 24, 5,0,1,0,spam,1

1351,1,1,0,0,0,spam,1

123.375,74,15,0,0.125,0,spam,1

122.800006,14,0,0,133333,0,spam,1

4.5,5,5,0,1,0,spam,1

0,2,0,0,1,0,spam,1

7.237903,532,248,0.000001,0.995968,0,spam,1

}

REFERENCES

[1] Jason Brownlee, "Artificial Immune Recognition Systems – A Review and Analysis", Technical Report, No 1-02, Centre for Intelligent Systems and Complex Processes, Faculty of Information and Communication Technologies, Swinburne University of Technology (SUT), 2005.

[2] Jason Brownlee, "Clonal Selection Theory and CLONALG – the clonal selection classification algorithm", Technical Report, No 2-01, Centre for Intelligent Systems and Complex Processes, Faculty of Information and Communication Technologies, Swinburne University of Technology (SUT), 2005.

[3] Jason Brownlee, "Immunos 81", Technical Report, No 3-01, Centre for Intelligent Systems and Complex Processes, Faculty of Information and Communication Technologies, Swinburne University of Technology (SUT), 2005.

[4] Jaber Karimpour, Ali A. Noroozi and Adeleh Abadi, "The Impact of Feature Selection on Web Spam Detection", *International Journal Intelligent Systems and Applications*, Vol. 4, No. 9, pp. 61-67, 2012.

[5] Miklós Erdélyi, András Garzó and András A. Benczúr, "Web Spam Classification: a Few Features Worth More", *Proceedings of the Joint WICOW/AIRWeb Workshop on Web Quality*, pp. 27-34, 2011.

[6] Guang-Gang Geng,, Chun-Heng Wang and Qiu-Dan Li, "Improving Web Spam Detection with Re-Extracted Features", *Proceedings of the 17th International Conference on World Wide Web*, pp.1119-1120, 2008.

[7] Shengen L, Xiaofei N, Peiqi L and Wang L, "Generating new features using genetic programming to detect link spam", *Proceedings of International Conference on Intelligent Computation Technology and Automation*, Vol. 1, pp. 135-138, 2011.

[8] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini and Sebastiano Vigna, "A Reference collection for Web Spam", *ACM SIGIR Forum*, Vol. 40, No. 2, pp. 11-14, 2006.

[9] www.cs.waikato.ac.nz/ml/weka/

[10] http://en.wikipedia.org/wiki/Targeted_projection_pursuit

[11] http://en.wikipedia.org/wiki/Formal_concept_analysis

[12] www.kdnuggets.com