DOI: 10.21917/ijsc.2016.0183

NLSDF FOR BOOSTING THE RECITAL OF WEB SPAMDEXING CLASSIFICATION

S.K. Jayanthi[1] and S. Sasikala[2]
[1]*Department of Computer Science, Vellalar College for Women, India*
E-mail: [1]sasi_sss123@rediff.com
[2]*Department of Computer Science, Hindusthan College of Arts and Science, India*
E-mail: [2]jayanthiskp@gmail.com

Abstract

Spamdexing is the art of black hat SEO. Features which are more influential for high rank and visibility are manipulated for the SEO task. The motivation behind the work is utilizing the state of the art Website optimization features to enhance the performance of spamdexing detection. Features which play a focal role in current SEO strategies show a significant deviation for spam and non-spam samples. This paper proposes 44 features named as NLSDF (New Link Spamdexing Detection Features). Social media creates an impact in search engine results ranking. Features pertaining to the social media were incorporated with the NLSDF features to boost the recital of the spamdexing classification. The NLSDF features with 44 attributes along with 5 social media features boost the classification performance of the WEBSPAM-UK 2007 dataset. The one tailed paired t-test with 95% confidence, performed on the AUC values of the learning models shows significance of the NLSDF.

Keywords:

Web Spam, Search Engine, SVM, Decision Table, HMM

1. INTRODUCTION

The inevitable role of the search engines are proved day by day whenever every user seeks for a piece of the information from the Web. Digital content flourishes at a lightning speed every second. The challenge of every content provider in Web is to get higher visibility in SERP for their domain. A survey report [1] indicates the importance of the results in first page. 91.5% of the traffic in Google is flooded to the first SERP. When moving from page one to two, the traffic dropped by 95%, and by 78% and 58% for the subsequent pages.

They also reveal a study for traffic share among the Websites based on their results position using a dataset drawn from May 21st to May 27th 2013. A website with the first position in the search results contributed to 33% of the traffic, compared to 18% for the second position. The data also shows that the first position on any page of results contributed to more traffic than the second position in each respective page (i.e. traffic from users dropped by 27%, 11.3%, and 5.4% from the first position to second position in page two, three, and four). The traffic dropped by 140% going from 10th to 11thposition and 86% going from 20th to 21st position. The drop in cumulative traffic moving from one page to another was even more significant.

Getting higher visibility would lead to many potential benefits. SEO task gains attraction at this point. SEO is the process of improving the visibility of a website or a web page in a search engine result page (SERP). SEO analyses the working method of ranking algorithm in search engines, factors concentrated while ranking and influential features of Website which improves rank for specific ranking algorithm. Along with that they also focus on the user profiling; which content user seeks in trending, what type of users seeks which type of information and class of targeted audience for specific search engines.

SEO is a task of manipulating content and hyperlinks of a Website. SEO techniques can be classified into two broad categories: white hat SEO, and black hat SEO. If a Website is subject to white hat SEO it results in genuine rank and visibility in SERP. In contrast, if it is subject to black hat SEO it gives higher than deserved rank and visibility in SERP. The search engines attempt to minimize the effect of the latter called spamdexing. Gyongyi and Garcia-Molina (2004) [2] discusses about the SEO in the web spam arena. The activity of some SEOs benefits the whole web community, as they help authors create well-structured, high-quality pages. However, most SEOs engage in practices that are called spamming.

For instance, there are optimizers who define spamming exclusively as increasing relevance for queries not related to the topics of the page. These SEOs endorse and practice techniques that have an impact on importance scores, to achieve what they call ethical web page positioning or optimization. According to them, all types of actions intended to boost ranking (either relevance, or importance, or both), without improving the true value of a page, are considered spamming.

WEBSPAM-UK 2007 dataset is the largest compiled collection with class labels. It is the most prominent dataset available in public to study the characteristics of the spamdexing, when this research has been started on 2010. Since the strategies in spamdexing is getting changed day by day, the pace of spamming changes more frequently. It is decided to collect features pertaining to SEO for selected samples in the dataset. Samples could be used to elucidate the structural traits and temporal amends of the spamdexing phenomenally.

Selecting the influential SEO features is the next task which involves an analysis in four renowned SEO sources [3]: SEOmoz, Searchengineland, Search engine journal and SEObook. Page-specific and domain-specific link based metrics play a vital role in optimizing a Website.

The page specific metrics considered as important in SEO are: keyword-focused anchor text from external links, external link popularity, diversity of link sources (links from many unique root domains), page-specific TrustRank, link popularity, focus of external link sources, keyword-focused anchor text from internal links, location in information architecture of the site, internal link popularity.

The site specific metrics which are of high importance in SEO are: trustworthiness of the domain based on link distance from trusted domains (TrustRank, Domain mozTrust), global link popularity of the domain (PageRank on the domain graph, Domain mozRank), link diversity of the domain (based on number of unique root domains linking to pages on this domain),

links from hubs/authorities in a given topic-specific neighborhood, temporal growth/shrinkage of links to the domain (the quantity/quality of links earned over time and the temporal distribution), links from domains with restricted access extensions (e.g. .edu, .gov, .mil, .ac.uk, etc.).

It is clear that Domain Trust/Authority acts as the dominant factor in the success of rankings at Google. A link strategy that positions a site as an authority, or a hub, in a web community is a powerful way to get attention of search engines [4]. According to a correlation study done by SEOmoz, a famous SEO concern, link metrics which has significance in visibility and visitability to a Website are: Domain authority, Page Authority, mozRank, mozTrust and link counts (external and internal) [3].

MozRank and PageRank scores are being used to show value on sites for sale in recent times. Page/Domain Authority represent SEOmoz's best ranking predictions and uses MozRank and MozTrust in their calculations. MozRank is a part of Page/Domain Authority mix which gives high granularity in spamdexing detection.

Cervino and Malae [5] conducted cyber metric analysis to analyze the correlation between applied metric indicators and accessibility problems in academic websites. They used the mozRank and domain authority as their focal indicators for determining the visibility for a Website. After analyzing the SEO sources available in Web, a set of features were demarcated. For naming convention, the additional features are collected, computed and compiled in the name NLSDF (New Link Spam Detection Features).

This paper is organized as follows: Section 2 discusses the related work in this problem. Section 3 describes 27 new features used for this work along with 17 computed features. Section 4 gives a brief about the suite of the classifiers used in this paper. Parameter settings of the classifiers are also briefed. Section 5 briefs the evaluation metrics and presents the results. Section 6 concludes the paper.

2. RELATED WORK

Erdelyi et al. [6] 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 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.

Benczur et al. [7] proposed a number of features based on the occurrence of the keywords that are either of high advertisement value or highly spammed. The new features are OCI (Online Commercial Intention), MindSet, Adwords, google Adsense, Pagecost. They are incorporated with WEBSPAM-UK2006 dataset. Newly included features improve the performance of the dataset by 3% accuracy.

Chung et al. [8] proposed new set of features including white score, spam score, relative trust; outgoing and incoming link related features, PageRank and hijacked score.

Kariampor et al. [9] 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. [10] 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.

Shen et al. [11] 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.

Jayanthi and Sasikala used genetic algorithm for Web spam classification [12]. Later utilized the Reptree based classifier for the same [13]. Naive bayes classifier is also proposed by the authors for the problem [14]. They also applied the Artificial Immune Recognition System based classifiers for the web spam problem. They proved that AIRS1 and AIRS2 Parallel are two methods which give best results when compared with pioneered literature [15].

3. NLSDF FEATURES

Social media has high correlation with the search engine ranking factors. 44 NLSDF features [16] are combined with the 5 influential social media SEO features. They are: Facebook shares, Facebook Likes, Number of Tweets, Google +1 and LinkedIn Shares. These data is collected and compiled for the existing NLSDF dataset

3.1 NLSDF BASE DATASET

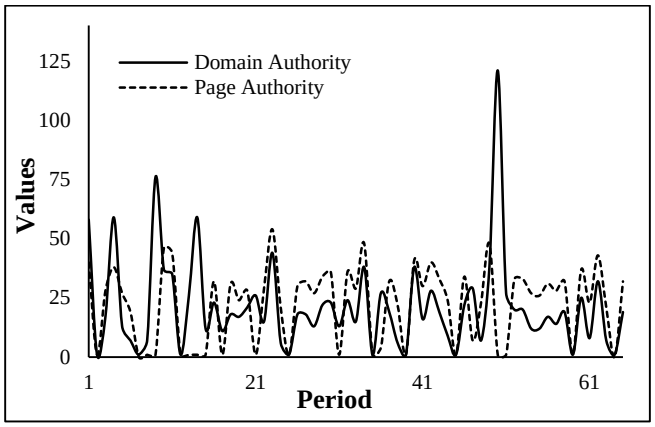
The commercial SEO features considered in NLSDF Base are enlisted below [16]:

- F1 Authority score of Domain
- F2 Authority score of the webpage
- F3 RD_Number of linking domains
- F4 Total number of anchor texts in website
- F5 SEOrank of the webpage
- F6 SEOtrust score of the webpage
- F7 Internal Links excluding 'Nofollow'
- F8 External Links excluding 'Nofollow'
- F9 Total number of internal links
- F10 Total number of external links

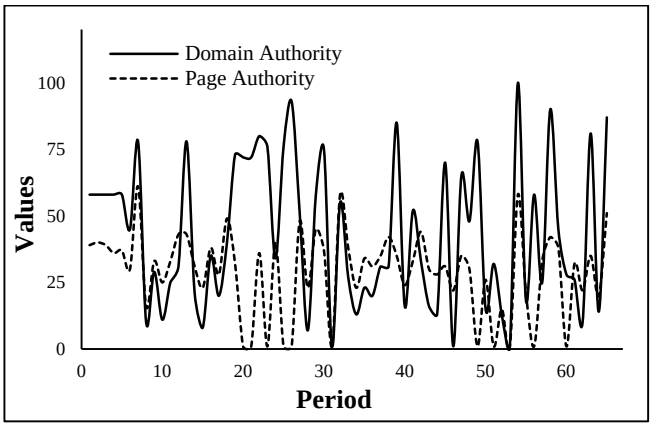
- F11 Cumulative total of the links in webpages
- F12 Linking RD excluding 'Nofollow'
- F13 Total number of linking RD
- F14 SD_SEOrank
- F15 SD_SEOTrust
- F16 SD_External Links excluding 'No-Follow'
- F17 SD_Total number of external links to SD
- F18 SD_Cumulative Total Links
- F19 SD_Linking RD excluding 'No-Follow'
- F20 SD_Total Linking Root Domains
- F21 RD_SEOrank
- F22 RD_SEOtrust
- F23 RD_External Links excluding 'No-Follow'
- F24 RD_Total number of external links
- F25 RD_Cumulative total links
- F26 RD_Linking Root Domains excluding 'No-Follow'
- F27 RD_Total Linking Root Domains

RD stands for the Root Domain and SD stands for the Sub Domain.

Corresponding values of these features for the websites listed in Base dataset is collected from various sources on web (Dmoz open directory and Google search engine) [17] [18].



(a) Non-Spam



(b) Spam Samples

Fig.1. Comparisons of Domain and Page Authority in (a) and (b)

The deviation between the domain and page authority values for spam and non-spam sample is depicted in Fig.1. It is evident that spam sample have higher deviation throughout the period because of the frequent content alterations.

The Fig.2 depicts the distribution of the TrustRank of HomePage (TrustRank_HP), mozTrust of HomePage (mozTrust_HP), MozTrust of Sub Domain (mozTrust_SD) and mozTrust of Root Domain (mozTrust_RD). Majority of samples values of the TrustRank lies in significantly low range for both spam and non-spam samples. mozRank related features in the graph shows the remarkable deviation and granularity between spam and non-spam samples. By introducing these features along with the existing features may thus improve the accuracy of the spamdexing classification.

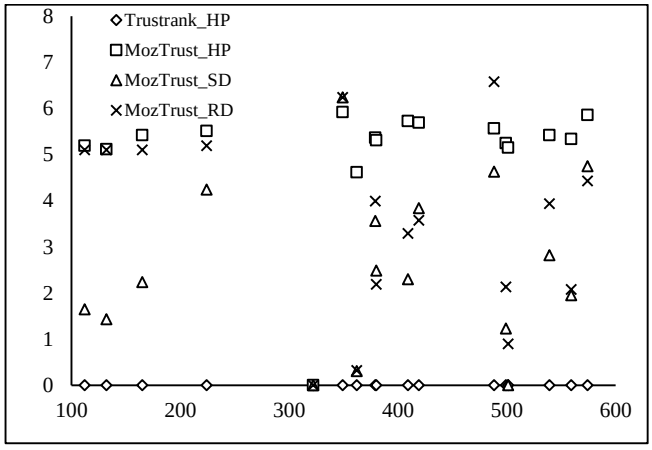


Fig.2. TrustRank and MozTrust Distribution in NLSDF

3.2 NLSDF DERIVED DATASET

The base 27 features (F1 to F27) were listed above and the NLSDF Derived 17 metrics (F28 to F44) which were used through this work is listed herewith [16]:

F28 Page Trust Score (PTS):

This feature calculates the Web page trust score with the help of the mozRank of the home page and the mozTrust of the homepage.

$$PTS = (HP\_mozRank) / ((HP\_mozRank + HP\_mozTrust))$$

F29 Sub Domain Trust Score (SDTS):

This feature calculates the sub domain trust score with the help of the mozRank of the sub domain and the mozTrust of the sub domain.

$$SDTS = (SD\_mozRank) / ((SD\_mozRank + SD\_mozTrust))$$

F30 Root Domain Trust Score (RDTS):

This feature calculates the root domain trust score with the help of the mozRank of the root domain and the mozTrust of the root domain.

$$RDTS = (RD\_mozRank) / ((RD\_mozRank + RD\_mozTrust))$$

F31 Cumulative Average Trust Score for Website (CTW):

This feature averages the trust scores calculated from home page (PTS), sub domain (SDTS) and root domain (RDTS).

$$CTW = (PTS + SDTS + RDTS) / 3$$

F32 Page Valid Links (HP_V_Links):

This feature finds the valid links present in a home page of a Website based on the total links (*HP_Tot_Links*), internal followed links (*HP_Int_FL*) and external followed links (*HP_Ext_FL*). The term valid refers to trusted and authority links. Number of broken links and nofollow links will be discarded.

$$HP\_V\_Links = HP\_Tot\_Links - (HP\_Int\_FL + HP\_Ext\_FL)$$

F33 Page Valid Linking Root Domain (HP_V_LRD):

This feature calculates the valid linking root domain links present in a home page of a Website. It deducts the total number of followed linking root domains present in a home page (*HP_FLRD*) from total number of links from the root domain (*HP_TLRD*).

$$HP\_V\_LRD = HP\_TLRD - HP\_FLRD$$

F34 Authority Score for Website (W_AScore):

This feature calculates the authority score for a Website based on the values obtained from page authority score (*P_AScore*) and domain authority score (*D_AScore*).

$$W\_AScore = (P\_AScore) / (D\_AScore)$$

F35 Rank Score based on Homepage Predictions (HP_SEORank):

This feature sum up the values of the mozRank values of the homepage (*HP_mozRank*), sub domain (*SD_mozRank*) and root domain (*RD_mozRank*).

$$HP\_SEORank = (HP\_mozRank + SD\_mozRank + RD\_mozRank)$$

F36 Trust Score based on Homepage Predictions (HP_SEOTrust):

This feature sum up the values of the mozTrust values of the homepage (*HP_mozTrust*), sub domain (*SD_mozTrust*) and root domain (*RD_mozTrust*).

$$HP\_SEOTrust = (HP\_mozTrust + SD\_mozTrust + RD\_mozTrust)$$

F37 SEO Spam Mass (SEO_SpamMass):

This feature calculates the amount of link spamming and figures out the number of links which were targeted with the intention of link spamming.

Gyongyi and Garcia-Molina proposed spam mass estimation with the help of PageRank and TrustRank values [19], whereas the proposed *SEO_SpamMass* feature finds the spam mass in search engine optimization perspective.

$$SEO\_SpamMass = (HP\_SEORank - HP\_SEOTrust) / (HP\_SEORank)$$

F38 Home Page Spam Mass (HP_SM):

This calculates the spam mass value of the home page with the help of mozRank (HP_mozRank) and mozTrust (HP_mozTrust) values of the home page.

$$HP\_SM = (HP\_mozRank – HP\_mozTrust)/(HP\_mozRank)$$

F39 Sub Domain Spam Mass (SD_SM):

This calculates the spam mass value of the sub domain with the help of mozRank (*SD_mozRank*) and mozTrust (*SD_mozTrust*) values.

$$SD\_SM = (SD\_mozRank – SD\_SEOTrust)/(SD\_SEORank)$$

F40 Root Domain Spam Mass (RD_SM):

This calculates the spam mass value of the root domain with the help of mozRank (*RD_mozRank*) and mozTrust (*RD_mozTrust*) values.

$$RD\_SM = (RD\_mozRank – RD\_mozTrust)/(RD\_mozRank)$$

F41 Average Spam Mass value for a Website (ASM_W):

This calculates the spam mass value for a Website based on the values of the *HP_SM, SD_SM* and *RD_SM*.

$$ASM\_W = HP\_SM + SD\_SM + RD\_SM$$

F42 Page Trust over Rank (PTR):

This feature calculates the trust over the rank of the Website home page using the mozRank and mozTrust values of the home page.

$$PTR = (HP\_mozTrust) / (HP\_mozRank)$$

F43 Sub Domain Trust over Rank (SDTR):

This feature calculates the trust over the rank of the sub domain using the mozRank and mozTrust values.

$$PTR = (HP\_\ mozTrust) / (HP\_mozRank)$$

F44 Root Domain Trust over Rank (RDTR):

This feature calculates the trust over the rank of the root domain using the mozRank and mozTrust values.

$$PTR = (HP\_mozTrust) / (HP\_mozRank)$$

3.3 NLSDF SOCIAL MEDIA IMPACT DATASET

F45 Facebook shares(SM_FS)

F46: Facebook Likes(SM_FL)

F47: Number of Tweets(SM_NT)

F48: Google +1 (SM_G)

F49: LinkedIn Shares (SM_LS)

Social media features impact the search results in prominent manner. Incorporating the metrics relevant to the social media helps to enhance the results of the spamdexing classification. NLSDF Base and Derived features [16] are deployed in this paper with social media features for boosting the recital of the search engine spamdexing classification.

4. PERFORMANCE EVALUATION OF NLSDF

Spamdexing is a binary classification problem. The deviation between the samples of spam and non-spam classes is learned by the classifier. Prediction is made for test sample based on the learned knowledge.

Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known [16]. MLT performance depends greatly on the characteristics of the data to be classified. There is no single technique that works best on all given problems (Global search heuristics, AIRA).

The primary objective of this experiment is leverage the performance of the NLSDF features. Results of single classifier could not be used to demarcate the overall performance of the features. The efficacy of the features could be revealed by subjecting to different methods apparently. Hence, in order to

evaluate the performance of NLSDF features a suite of MLT were experimented. Determining a suitable MLT for a given problem is however still challenging.

Caruana et al. [20] performed empirical evaluation on ten machine learning algorithms which gives interesting insights. Wu et al. [21] performed an extensive survey to determine top 10 algorithms in data mining. K. Dnuggets (2011) [22] suggests the best performing MLT based on a poll conducted on November 2011.

Silva et al. (2012) [6] evaluated the WEBSPAM-UK datasets (2006 and 2007) on different machine learners. Their objective is to find the best performing method on the dataset whereas the objective of this work is to assess the performance of the NLSDF datasets. They conclude that AdaBoost and Bagging Performs well on the datasets. Methods for further instigation were selected after having a review of literature.

Short overviews of these techniques are discussed in subsequent part of the section. MLT predicts the results based on the link features and SEO features which characterize the samples. These features encode samples in very high dimensional feature vectors.

The high dimensionality of these feature vectors poses certain challenges for classification. Though only a subset of the generated features may correlate with spamdexing detection, it is not known in advance which features are relevant. Feature selection can be applied in order to resolve this [15].

4.1 DATASET

Search Engine Optimization (SEO) features are proposed for web spamdexing detection in this work. Four datasets used in the experiments are: *Baseline* (WEBSPAM-UK 2007), *PDS*–1 ($NLSDF_{SEOBASE}$), *PDS*-2 ($NLSDF_{SEOCOMP}$) and *Baseline*+*PDS*-2 (WEBSPAM-UK 2007+ $NLSDF_{SEOCOMP}$). In *Baseline* dataset there are 44 attributes. In *PDS*-1, 27 attributes are introduced based on the SEO of a Website. In *PDS*-2, along with the 27 attributes of *PDS*-1 another 17 computed attributes are proposed. A total of 44 attributes will be present in the *PDS*-2. In *Baseline*+*PDS*-2, a total of 93 attributes from *Baseline* and *PDS-2* datasets along with the 5 social media features will be experimented.

Features and classifiers are the two pillars of the spamdexing detection with MLT. This paper explicates a new set of classifiers and features to accomplish optimal performance in WLS detection. WEBSPAM-UK 2007 dataset has been considered as the *baseline*. It contains 3998 feature values of samples with 222 spam hosts and 3776 non-spam hosts. The distributions of samples are of 5% spam and 95% non-spam. A minimal subset of the baseline dataset is considered in this work to demonstrate the effectiveness of the proposed features. The 3776 non-spam samples were divided into 10 buckets.

First 8 buckets each with 377 samples and the last two buckets has 380 samples. Spam samples are gathered as a single bucket and arranged in *high_to_low* assessment score order. Non-spam samples in each bucket are arranged in *low_to_high* assessment score order. In each bucket, top 50 samples were selected for this work. As a result, 500 non-spam samples and 222 spam samples were considered from the *baseline* dataset. Thus, strategic sampling is performed on non-spam samples.

A total of 18% samples from *baseline* dataset are selected for further process with a distribution of 30% spam and 70% non-spam. The reason to select all spam samples is to better characterize the features of spamdexing in state of the art scenario. Hence, task of collecting and computing the feature values needs a huge team effort; this work selects a set of 722 samples.

For each sample 27 feature values has to be collected and 17 features has to be computed. 19,494 values for the proposed features are collected from various sources in Web for compiling $NLSDF_{SEOBASE}$. 12,274 values are computed for $NLSDF_{SEOCOMP}$ and these 44 feature values are experimented along with the selected samples from *baseline and Social media samples*. Data values for the 27 aforesaid features are collected from the Open Site Explorer (OSE) - SEOmoz website (OSE 2013), Yahoo and Majestic SEO. $NLSDF_{SEOBASE}$ values are collected and incorporated in the base WEBSPAM-UK 2007 dataset Data values are collected from March 2012 to April 2012. $NLSDF_{SEOCOMP}$ is calculated and included. Performance of the new feature inclusion is tested against the base dataset with machine learning techniques. Results of experiments are discussed in the subsequent section of the Paper.

4.2 TEST METHODS

Experiments are carried out with the classifiers and datasets. For each dataset used in the experiments, perform MLT over 10 fold cross-validation where the entire data is utilized for training and testing. Experiments ran on a machine with 2 dual-core 2.33 GHz Pentium IV processors with 4 GB memory. Weka machine learning toolkit is used for the experiment. A short overview of methods used for the experiment is given.

4.2.1 AdaBoost:

The AdaBoost algorithm [14] is one of the most important ensemble methods, since it has solid theoretical foundation, very accurate prediction, great simplicity, and wide and successful applications. Given base algorithm and training dataset, AdaBoost updates the weights of the samples for specific rounds and final model is derived by weight majority voting of the base learners, where the weights of the learners are determined during the training process. In practice, the base learning algorithm may be a learning algorithm which can use weighted training examples directly; otherwise the weights can be exploited by sampling the training examples according to the weight distribution [21].

4.2.2 Support Vector Machine (SVM):

SVM is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. Support vector machine calibrated with Platt's method is remarkably effective at obtaining excellent performance on the probability metrics from learning algorithms that performed well on the ordering metrics. SVM calibrated with Platt's method is used for experiments with polynomial kernel. The SVM parameters are selected based on the suggestions given by Rich [20].

4.2.3 Decision Table (DT):

Decision Table algorithm build simple decision table majority classifier [24]. It summarizes the dataset containing the same number of attributes as the original dataset. Then, a new data item is assigned a category by finding the line in the decision

table that matches the non-class values of the data item. Decision Table employs the wrapper method to find a good subset of attributes for inclusion in the table. By eliminating attributes that contribute little or nothing to a model of the dataset, the algorithm reduces the likelihood of over-fitting and creates a smaller and condensed decision table [25]. Best-first search is adopted which searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility and search termination is set to 5 iterations.

4.2.4 Classification and Regression Tree (CART):

CART is a binary recursive partitioning procedure capable of processing continuous and nominal attributes both as targets and predictors. Trees are grown to a maximal size without the use of a stopping rule and then pruned back (essentially split by split) to the root via cost-complexity pruning. The next split to be pruned is the one contributing least to the continuous and nominal attributes both as targets and predictors [21]. The maximum tree depth is set to -1 for no restriction. The minimum total weight of the instances in a leaf is set to 2. Pruning is performed on the trees with one fold; the value for growing the rules is set to 3.

4.2.5 Ensemble Selection (ES):

Ensemble is a collection of models whose predictions are combined by weighted averaging or voting. The ensemble selection strategy adopted in this work is proposed [20]. It starts with empty ensemble; add model in the library to the ensemble which maximizes the performance to the specified metric on a validation set. The process ran for fixed number of iterations and nested set of ensemble which returns maximum performance. Greedy sort initialization is set to true for stop adding models when performance degrades. Hill climbing iterations for the ensemble selection algorithm is set to 100. Accuracy metric is used to optimize the chosen ensemble. Model ratio is set to 0.5, it is the ratio of library models that will be randomly chosen for each iteration. Number of model bags used in the ensemble selection algorithm is set to 10.

4.2.6 Expectation-Maximization (EM):

EM is a popular iterative refinement algorithm that can be used for finding the parameter estimates. Each object is assigned to the cluster based on the weight representing the probability of membership. It starts with initial estimate and iteratively rescores the object against the mixture density produced by the parameter vector [26]. The number of clusters is set to 2 and initial seed set value is 100.

4.2.7 Self-Organizing Maps (SOM):

SOM is a type of artificial neural network which implements Kohonen's Self Organizing Map algorithm for unsupervised clustering. It is a two layered network, first one is input layer and second one is the competitive layer. All the nodes in the competitive layer compare the inputs with their weights and compete with each other to become the winning unit having the lowest difference [27].

Calculating statistics for each cluster after training is set to true. Number of epochs in convergence phase is set to 100. The height and width of lattice is set to 2. Attribute normalization is adopted and the number of epochs in ordering phase is set to 2000.

4.2.8 DBSCAN:

DBSCAN is a density based spatial clustering of applications with noise; it finds the number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN requires two parameters: ε(eps) and the minimum number of points required to form a dense region (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started [28].

4.2.9 K-Means Cluster:

K-means is the simplest most popular classical clustering method. It uses K-clusters represented by the mean of the object (called Centroid). The method is a search problem where the aim is to find optimal clusters, given the number of clusters and seeds. Euclidean distance is used with two cluster and 100 iteration settings [29].

4.2.10 Learning Vector Quantization (LVQ):

LVQ is a special case of neural network related to k-Nearest Neighbor (k-NN) algorithm proposed by Teuvo Kohonen. An LVQ system is represented by prototypes $W = (w(i),...,w(n))$ which are defined in the feature space of observed data. For each data point, the winner prototype which is closest to the input according to a given distance measure is determined. The position of this winner prototype is then adapted, i.e. the winner is moved closer if it correctly classifies the data point or moved away if it classifies the data point incorrectly. An advantage of LVQ is that it creates prototypes that are easy to interpret for experts in the respective application domain [30].

5. TEST RESULTS

The Table.1 elucidates the performance comparison of the four datasets. In supervised learning models, F-Value of *Baseline+PDS-2* dataset show radical improvement in Decision Table and Ensemble Selection when compared with *Baseline*. F-Value of *Baseline+PDS-2* improves by 51% in Decision Table and in Ensemble Selection it improves by 50%. AdaBoost show 10% and CART show 5% improvement with the proposed NLSDF dataset Accuracy also seems to be better with the *Baseline+PDS-2* dataset for four supervised learning models.

PDS-1 and *PDS-2* models offer moderate performance in spamdexing classification with supervised learning models. So it is evident that when the proposed NLSDF features combined with WEBSPAM-UK 2007 features (*Baseline+PDS-2*), the performance of the supervised learning models improves in noteworthy manner. The mean accuracy value of supervised learning models for the *Baseline+PDS-2* dataset is 0.915 and for the *Baseline*, it is 0.692. NLSDF features improve the *Baseline* dataset accuracy by 22%, which is momentous.

In unsupervised learning models, SOM show 14% improvement for *Baseline+PDS-2* and K-Means clustering has 12% improvement. DBSCAN has 4% and LVQ has 2% improvement in F-Values respectively, whereas in EM the performance mortifies by 34%. Consequently, the mean accuracy value of the unsupervised learning models for *Baseline+PDS-2* mortifies by 7% when compared with the *Baseline*. Accuracy also seems to have the same kind of performance as of like F-Value. The PDS-1 and PDS-2 datasets show moderate performance in

unsupervised learning models. It is evident that *Baseline+PDS-2* dataset performs well supervised learning methods.

The Fig.3 shows the Mean Accuracy comparison of the supervised and unsupervised learning models for the *Baseline* and *Baseline + PDS-2* datasets. The black color bars indicate negative performance. Seven models substantiate positive performance whereas, three models indicate negative performance. Mean Accuracy comparison plotted in Fig.3 gives the insight of suitability of the learning models for spamdexing classification problem. In supervised learning models, SVM performs well, followed by AdaBoost, Decision Table, Ensemble Selection and CART respectively. In supervised learning models, LVQ performs well, followed by SOM, EM, K-Means and DBSCAN respectively. The one tailed paired t-test with 95% confidence, performed on the AUC values of the learning models shows significance.

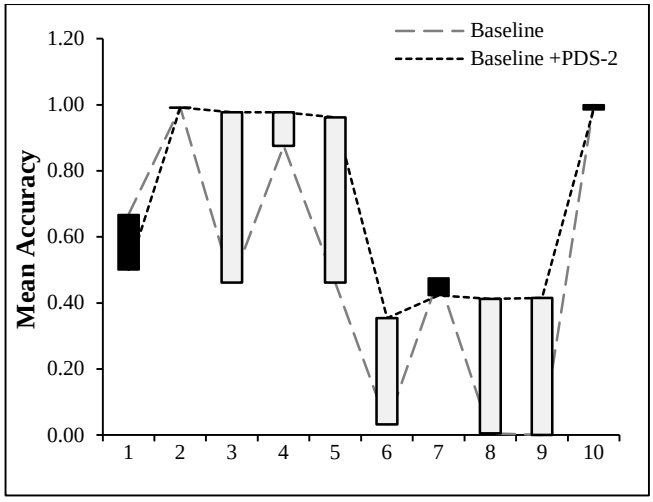


Fig.3. Mean Accuracy Comparison of Learning Models

Table.1. Test Results

Unsupervised Learning Models						Supervised Learning Models				
Method	**Precision**	**Recall**	**F-Value**	**Accuracy**		**Method**	**Precision**	**Recall**	**F-Value**	**Accuracy**
Baseline										
EM	1.000	0.969	0.984	0.985		AdaBoost	0.833	0.923	0.876	0.869
SOM	0.525	0.477	0.500	0.523		CART	0.361	0.200	0.257	0.423
DBSCAN	0.474	0.138	0.214	0.492		SVM	1.000	0.985	0.992	0.992
K-Means	0.500	0.123	0.198	0.500		DT	0.462	0.462	0.462	0.462
LVQ	0.940	0.969	0.955	0.954		ES	0.462	0.462	0.462	0.462
PDS-1										
EM	0.655	0.585	0.618	0.624		AdaBoost	0.717	0.662	0.688	0.700
SOM	0.361	0.200	0.257	0.423		CART	0.500	0.108	0.177	0.500
DBSCAN	0.500	0.108	0.177	0.500		SVM	0.897	0.538	0.673	0.738
K-Means	0.500	0.062	0.110	0.500		DT	0.712	0.569	0.632	0.669
LVQ	0.600	0.738	0.662	0.623		ES	0.709	0.601	0.650	0.677
PDS-2										
EM	0.531	0.785	0.634	0.546		AdaBoost	0.685	0.569	0.622	0.654
SOM	0.537	0.785	0.638	0.554		CART	0.533	0.123	0.200	0.508
DBSCAN	0.533	0.123	0.200	0.508		SVM	0.796	0.662	0.723	0.746
K-Means	0.538	0.108	0.179	0.508		DT	0.667	0.615	0.640	0.654
LVQ	0.608	0.692	0.647	0.623		ES	0.672	0.631	0.651	0.662
Baseline+PDS-2										
EM	0.537	0.785	0.638	0.554		AdaBoost	0.970	0.985	0.977	0.977
SOM	0.543	0.785	0.642	0.562		CART	0.533	0.123	0.200	0.508
DBSCAN	0.524	0.169	0.256	0.508		SVM	1.000	0.985	0.992	0.992
K-Means	0.583	0.215	0.315	0.531		DT	0.956	1.000	0.977	0.977
LVQ	0.970	0.985	0.977	0.977		ES	0.929	1.000	0.963	0.962

6. CONCLUSION

This paper advocates the problem of detecting spamdexing using machine learning techniques over website features and impact of social media features. It is evident that, the social media features improve the accuracy of spamdexing classification. In this paper, only link based features pertaining to SEO and social media are considered and hence it cannot detect the content based spam. When both features are combined then it could be possible to achieve more accurate results and this will be the future scope of the research.

REFERENCES

[1] The Value of Google Result Positioning, Available at: https://chitika.com/google-positioning-value.

[2] Z. Gyongyi and H.G. Molina, "Web Spam Taxonomy", *Proceedings of 1st International Workshop on Adversarial Information Retrieval on the Web*, pp. 39-47, 2004.

[3] Rand Fishkin, "The Science of Ranking Correlations: How Does PageRank Perform", Available at: https://moz.com/blog/the-science-of-ranking-correlations.

[4] Sona Makulova, "A Qualitative Analysis of the Factors Influencing Findability of the Websites of Slovak Libraries", *Proceedings of 10th Symposim of Polish Society for Information Science*, pp. 1-8, 2009

[5] Sanchez Cervillo, Miguel and Enrique Orduna-Malea, "Influence Of WCAG Rules on Academic Websites Rankings: A Correlation Study Between Accessibility And Quantitative Webometrics", *Proceedings of 2nd International AEGIS Conference and Final Workshop*, pp. 196-203, 2011

[6] Miklos Erdelyi, Andras Garzo and Andras A. Benczur, "Web Spam Classification: A Few Features Worth More", *Proceedings of Workshop on Web Quality*, pp. 27-34, 2011.

[7] A. Benczur, I. Bıro, K. Csalogany and T. Sarlos, "Web spam Detection via Commercial Intent Analysis", *Proceedings of 3rd International Workshop on Adversarial Information Retrieval on the Web*, pp. 89-92, 2007.

[8] Young Joo Chung, Masashi Toyoda and Masaru Kitsuregawa, "Identifying Spam Link Generators for Monitoring Emerging Web Spam", *Proceedings of 4th Workshop on Information Credibility*, pp. 51-58, 2010.

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

[10] Guanggang Geng, Chunheng Wang and Qiudan Li, "Improving Web Spam Detection with Re-Extracted Features", *Proceedings of the 17th International Conference on World Wide Web*, pp. 1119-1120, 2008.

[11] Guoyang Shen, Bin Gao, Tie Yan Liu, Guang Feng, Shiji Song and Hang Li, "Detecting Link Spam Using Temporal Information", *Proceedings of 6th International Conference on Data Mining*, pp. 1049-1053, 2006.

[12] S.K. Jayanthi and S. Sasikala, "Wespact Detection of Web Spamdexing with Decision Trees in GA Perspective", *Proceedings of International Conference on Pattern Recognition, Informatics and Medical Engineering*, pp. 381-386, 2012.

[13] S.K. Jayanthi and S. Sasikala, "Reptree Classifier for Identifying Link Spam in Web Search Engines", *ICTACT Journal of Soft Computing*, Vol. 3, No. 2, pp. 498-505, 2013.

[14] S.K. Jayanthi and S. Sasikala, "Naive Bayesian Classifier and PCA for Web Link Spam Classification", *Georgian Electronic and Scientific Journal: Computer Science and Telecommunications*, Vol. 41, No. 1, pp. 3-15, 2014.

[15] S.K. Jayanthi and S. Sasikala, "Web Link Spam Identification Inspired By Artificial Immune System and the Impact of TPP-FCA Feature Selection on Spam Classification", *ICTACT Journal of Soft Computing*, Vol. 4, No. 1, pp. 633-644, 2013.

[16] S.K. Jayanthi, S. Sasikala and J.P. Vishnupriya, "Comprehensive Evaluation of Machine Learning Techniques and Novel Features for Web Link Spamdexing Detection", *International Journal of Research in Science*, Vol. 1, No. 2, pp. 98-109, 2014.

[17] Iwebtool, Accessed on 2012, Available at: http://www.iwebtool.com/pagerank_prediction.

[18] Dmoz, Available at: http://www.dmoz.org/, Accessed on 2012.

[19] Zoltan Gyongyi, Pavel Berkhin, Hector Garcia Molina and Jan Pedersen, "Link Spam Detection based on Mass Estimation", *Proceedings of 32nd International Conference on Very Large Data Bases*, pp. 439-450, 2006.

[20] Rich Caruana and Alexandru Niculescu-Mizil, 2006,"An Empirical Comparison of Supervised Learning Algorithms", *Proceedings of 23rd International Conference on Machine Learning*, pp. 161-168, 2006

[21] X. Wu *et al.*, "Top 10 Algorithms in Data Mining", *Knowledge and Information Systems*, Vol. 14, No. 1, pp. 1-37, 2008.

[22] Algorithms for Data Analysis/Data Mining, Available at: http://www.kdnuggets.com/polls/2011/algorithms-analytics-data-mining.html.

[23] R.M. Silva, A. Yamakami and T.A. Almeida, "An Analysis of Machine Learning Methods for Spam Host Detection", *Proceedings of 11th International Conference on Machine Learning and Applications*, Vol. 2, pp. 227-232, 2012.

[24] Ron Kohavi, "The Power of Decision Tables", *Proceedings of 8th European Conference on Machine Learning*, pp. 174-189, 1995.

[25] Induction Algorithm, Available at: https:/mydatamining.wordpress.com/tag/induction-algorithms, Accessed on 2008.

[26] Jiawei Han, Micheline Kamber and Jian Pei, "*Data Mining: Concepts and Techniques*", 3rd Edition, Morgan Kaufmann, 2011.

[27] Sankar K. Pal and Pabitra Mitra, "*Pattern Recognition Algorithms for Data Mining*", 1st Edition, CRC Press, 2004.

[28] G.K. Gupta, "*Introduction to Data Mining with Case Studies*", 2nd Edition, Prentice-Hall, 2011.

[29] Oswal Sangita, "An Improved K Means Clustering Approach for Teaching Evaluation", *Proceedings of International Conference on Advances in Computing, Communication and Control*, pp. 108-115, 2011.

[30] Learning Vector Quantization, Available at: https://en.wikipedia.org/wiki/Learning_vector_quantization