IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-5 E-ISSN: 2347-2693

Spam Detection on Social Media Text

G. Jain[1*], Manisha[2], B. Agarwal[3]

[1]Department of Computer Science, Banasthali University, Banasthali, India
[2] Department of Computer Science, Banasthali University, Banasthali, India
[3]Department of Computer Science and Engineering, SKIT, Rajasthan University, India

**Corresponding Author: jain.gauri@gmail.com, Tel.: +91-98186-63662*

Online Available at: www.ijcseonline.org

Received: 20/Apr/2017, Revised: 28/Apr/2017, Accepted: 21/May/2017, Published: 30/May/2017

Abstract**—** Communication has become stronger due to exponential increase in the usage of social media in the last few years. People use them for communicating with friends, finding new friends, updating any important activities of their life, etc. Among different types of social media, most important are social networking sites and mobile networks. Due to their growing popularity and deep reach, these mediums are infiltrated with huge Vol.of spam messages. In this paper, we have discussed 5 traditional machine learning techniques for detecting spam in the short text messages on two datasets: SMS Spam Collection dataset taken from UCI Repository and Twitter dataset. Twitter dataset is compiled by crawling the public live tweets using Twitter API. The BoW with TF and TF-IDF weighing schemes is used for feature selection. The performance of various classifiers is evaluated with the help of metrics like precision, recall, accuracy and F1 score. The results show that the Random Forest gave highest accuracy with 100 estimators.

Keywords**—**Spam Detection, machine learning, Traditional classifiers, Twitter spam, SMS spam, Text Classification

I. INTRODUCTION

Spam refers to the irrelevant or unsolicited messages sent over the network with the sole intention of attracting the attention of a large number of people [1]. Spam may or may not be harmful to the intended person. It might range from just a funny text message to a deadly virus that may corrupt the entire machine or a code written to steal all the information on your machine. Initially, the spam started spreading with email, but with the increase in the use of the Internet and the advent of social media [2], they started to spread like an epidemic.

According to a technical report by Ferris Research Group [3], it is stated that these types of mails occupy a chunk of bandwidth and storage space with the user wasting their precious time and energy in avoiding these types of mails. This has resulted in the financial strain on organizations, increased requirement of storage, spreading of offensive material like pornographic content and above all it violates the privacy of the people at the receiving end. Other mediums of spam are social networking sites, spam blogs, etc. which are used to send/ receive messages and the SMS which carry spam over mobile networks. The increasing awareness about the email spam has decreased the return rate drastically, therefore traditional spammers are now using mobile and Internet technologies as a spam medium. With the widespread availability of smart phone, there is an increase in the Vol.of data exchanged over the network. SMS is a very cost effective method used for exchanging messages and therefore these can be used to send to the users individually. It has a higher response rate as compared to email spam. Apart from emails, and SMS [4], social networking like Twitter [5], Facebook, instant messenger like WhatsApp etc. are also contributing to a major chunk of spam over the network.

Spam detection is a tedious task and in the absence of automatic measure for filtering of message, the task of spam filtering is taken up with the person at the receiving end. One of the measures for spam protection is to include Ad hoc classifiers. These are the classifiers are applicable in response to a particular type of spam or to restrict spam messages from a particular source. Examples of these types of classifiers include blocking the incoming messages from a particular source by the email client knowing that the sender's address is in the blacklist.

Similar to Ad hoc classifiers, there is a rule based filtering, with the difference that rules are more formally written and can be deployed to a wide area of clients. A set of pre-defined rules are applied to an incoming messages and the message is marked as spam if the score of the test exceeds the threshold specified. Survey of anti–spam tools are provided in [6, 7]. Many companies have additional checks in the form of white–listing, black–listing [8, 9] and grey–listing [10]. However, the success of these methods is limited and they need to be combined with automatic machine learning methods in order to get fairly good results. Machine learning algorithms comes under the category of content based classification technique since the properties and features are extracted from the text of the message. Some most common classifiers used for spam detection are SVMs, Naïve Bayes, Artificial Neural Network, and Random Forests. These classifiers need a way to extract features from the text. The most common model for feature extraction is Bag – of – words (BoW). There are different weighing schemes in BoW model like Term Frequency (TF), Inverse Document Frequency (TF – IDF) etc., but all of them uses the token frequency in some form. Agarwal [11] has defined various composite features like information gain (IG), minimum redundancy maximum relevancy (murmur) besides new bi – tagged features like POS, etc. Agarwal [12] has used also used common sense knowledge using ConceptNet for feature extraction.

In the above mentioned classifiers, the most effective and a simple statistical classifier is a Naïve Bayes among other classifiers [13]. It is most widely used and researched. It assumes that the features extracted from the word vector are independent of each other [14, 15]. Much of the work is done in the area of spam detection using Naïve Bayes. Yang [16] proposed Naïve Bayes classifier ensemble based on bagging which improved the accuracy of the classifier. Kim [17] experimented with different no. of features used for spam classification using the same algorithm. Androutsopoulos [18] also performed a comparison between naïve bayes and key –word based spam filtering on social bookmarking system and concluded that the performance of Naïve Bayes is better amongst the two. Almeida [19] used different techniques like document frequency, information gain, etc. for term selection and used it with four different versions of Naïve Bayes for spam filtering. He concluded that Boolean attributes perform better than others and MV Bernoulli performs best with this technique.

Apart from Naïve Bayes, another very popular traditional classifier is a Support Vector Machine (SVM) [20]. Like Naïve Bayes, SVM is also used for detection of spams from various social media like Twitter [21], Blogs [22], etc. There are various variations introduced to further enhance the performance of SVM classifier. For e.g. Wang [23] proposed GA-SVM algorithm in which genetic algorithm used for feature selection and SVM for the classification of spams and its performance was better than SVM. Tseng [24] created an algorithm that gave an incremental support to SVM by extracting features from the users in the network. This model proved to be effective for the detection of spam on email. Functional spam detection is done with the help of temporal position in the multi – dimensional space.

Other than SVM, another functional classifier is k – NN [25]. It has also given good results in the area of spam detection. There are many other researchers who used KNN for spam detection in different applications [26, 27, 28]. Artificial Neural Network (ANN) has also shown promising results in the area of spam detection. Sabri [29] used ANN for spam detection in which the useless input layers could be changed over a period of time with the useful one. Silva [30] compared different types of ANN like MLP, SOM, Levenberg-Marquardt algorithm, RBF for content based spam detection, concluded that some of them have high potential. Ensemble methods have proven their capability as a classifier in the field of spam detection. One of the highly efficient classifier is random forest. DeBarr [31] used clustering along with Random Forest for spam classification. Several researchers gave comparison between the above discussed classifiers in literatures [32, 33, 34].

The rest of the paper is organized as follows: Section II discusses various traditional machine learning algorithms. Experiments and results are given in Section III and finally Section IV concludes the results and findings.

II. TRADITIONAL MACHINE LEARNING CLASSIFIERS

A. Support Vector Machine (SVM)

Support vector machine (SVM) [20] is a non – probabilistic supervised learning classifier that assigns a class label to the test data. The training text data are represented in the form of n–dimensional vector that consists of a list of numbers which are represented as points in n – dimensional space. SVM finds a suitable (n – 1)–dimensional hyperplane that separates different classes of data objects. There might be many hyperplanes which act as classifier, but the hyperplane is chosen such that the distance between the classes on each side of data points is maximized.

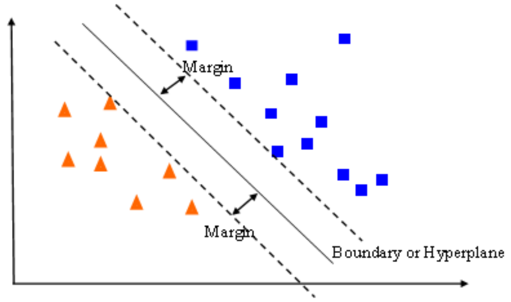


Figure 1. Support Vector Machine

Figure 1 shows the hyperplane that lies half way in between them is known as maximum – margin hyperplane. Two parallel hyperplanes have to be found that separate the two classes by the maximum distance between them. The region bounded by these two hyperplanes is known as margins. Classification of SVM is based on features extracted from the training text. These features can be extracted using variety of techniques like Bag-of-words (BoW), chi-square etc.

B. Naïve Bayes Classifier (NB)

Naïve Bayes (NB) [35] classifier is one of the oldest and most effective machine learning method. The main reason for its popularity is its high performance while maintaining its simplicity. It is a probabilistic classifier which classifies the data instance according to the frequency/ probability of the feature vector while having an unrealistic assumption of feature independence. NB classifier is a supervised statistical learning algorithm based on Bayes' Theorem given by Thomas Bayes (1702 – 1761).

$$\boldsymbol{P}(\boldsymbol{C}|\boldsymbol{D}) = \frac{\boldsymbol{P}(\boldsymbol{D}|\boldsymbol{C}) * \boldsymbol{P}(\boldsymbol{C})}{\boldsymbol{P}(\boldsymbol{D})} \qquad (1)$$

where *C* and *D* are events and *P(D) ≠ 0.*

P(D) and *P(C)* are the prior probabilities of observing D and C without regard to each other.

P(C | D), a conditional probability or posterior probability, is the probability of observing event *C* given that *D* is true.

P(D | C) is the probability of observing event *D* given that *C* is true.

In the above rule, one conditional probability of one event is considered while in NB classifier, the probability of an event is computed based on many different events or features in case of text classification.

C. Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) were initially developed to model the working of the human brain. It is a computational model of the biological neural network and takes inspiration from biological working of specialized cell called Neurons. Theoretically speaking ANN is able to emulate the learning process of concepts by human brain. The most basic unit of ANN is known as neuron or a node. The Figure 2 shows a single neuron. The input is received through other nodes connected with the weighted edges or connections. The weights on the connections are dependent on the relative importance of the inputs. The output is connected as the function of sum of weighted inputs.

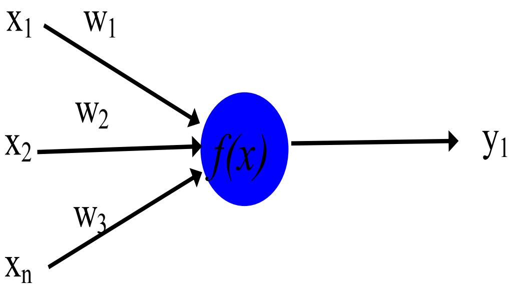


Figure 2. A Single Neuron

The function *f(x)* is known as activation function and the output y_1 is calculated as shown in Eq. (2):

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}_1 = \boldsymbol{x}_1 \bullet \boldsymbol{w}_1 + \boldsymbol{x}_2 \bullet \boldsymbol{w}_2 + \boldsymbol{x}_3 \bullet \boldsymbol{w}_3 \qquad (2)$$

A typical ANN is described below:

Architecture of ANN:

Figure 3 shows a basic artificial neural network, which consists of three layers: An input layer, a hidden Layer and an output layer. The input layer consists of neurons, which receives input from several other neurons and are connected with hidden layer via edges or synapses. These synapses stores the values called weights and help in manipulating the sending the output to the output layer. The connection between the neurons is dependent on the weight value, therefore, the node having more weight passes more information. Mathematically, ANN is directed graph *G* with an ordered 2-tuple *(V, E)* consisting of a set *V* of vertices and E of edges with vertices *V= {1,2,..,n}* and arcs *A= {<i,j> | i≥1,j≤ n}*

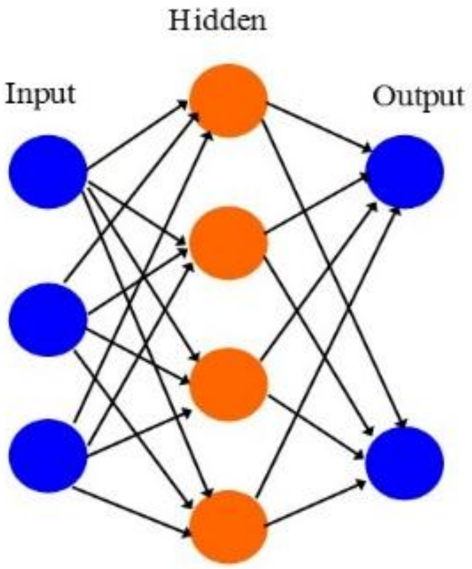


Figure 3. An Artificial Neural Network

D. k-Nearest Neighbor (KNN)

k – Nearest Neighbor (KNN) is one of the non – parametric supervised learning algorithm that gives fairly good results regardless of its simplicity. The main feature of KNN is that it is independent of the assumptions on the underlying data distribution, since our day to day real world data doesn't follow typical theoretical assumptions like linearly separable, Gaussian mixtures etc. This method is a lazy learning method where the generalization of the data is delayed until the query is placed rather than the generalization with the training data before actually receiving the query. Therefore, training phase is fast in comparison to other popular supervised learning classifiers which consists of storing the feature vector and class labels of the training samples, but a slow and costly testing phase in terms of time and memory. KNN approximates the target function depending on the individual query which results in a simultaneous solution of multiple problems and it also helps in dealing with variations in the problem domain as well.

In KNN, classifier works on the basis of k number of neighbors in its vicinity. The test object is assigned the label of the class of the most frequently occurring object among the *k* training samples nearest to the test object. If *k = 1* then the test object is assigned the class of its nearest neighbor.

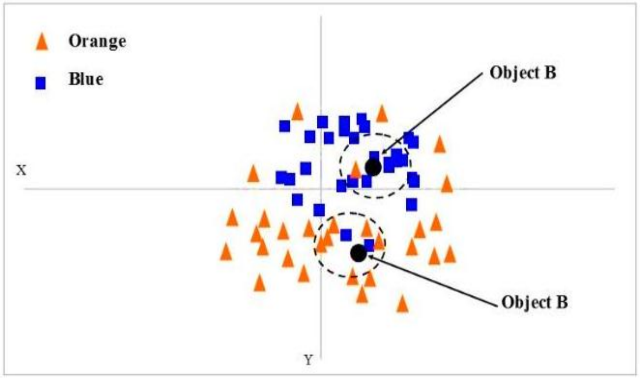


Figure 4. k – Nearest Neighbor Classifier

In the Figure 4, object *A* will be assigned the class label orange as the majority of its neighboring object (dashed circle) are orange and object *B* would be assigned class label blue. Some of the commonly used measures to calculate the distance between the test object and the various training object are Euclidean distance and Manhattan distance:

E. Random Forest (RF)

Random Forest [36] uses an ensemble methodology to build a classifier model. The ensemble technique integrates multiple models for improvement in the final prediction. In case of random forest, various decision trees are combined together so as to produce better results than a single tree. The dataset is divided into sub – sets repeatedly and the decision trees are constructed on the basis of various combinations of variables. Thus, huge number of trees are constructed by using various combinations and order of variables. The combination of randomly generated trees forms a forest, in the decision trees each node is split using the best set of variables which are randomly chosen at that node. This has an advantage over deep trees, as they tend to overfit during the training as go deep in order to reduce the error which increases the test error.

III. EXERIMENTAL SETUP

A. Preprocessing

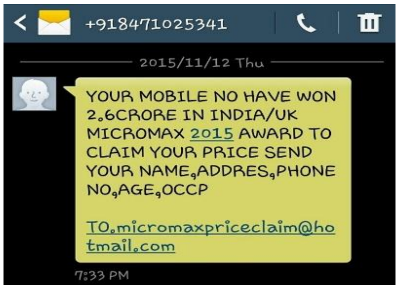
The natural language used in social media text is not structured and does not follow the language rules. Therefore, initial analysis and pre – processing is required for effective feature selection and classification. Pre - processing is one of the critical tasks in the area of text classification or Natural Language Processing (NLP). This is because the raw data from the source is generally incomplete, inconsistent or noisy and it requires cleaning and to be bought in the form where it can be used in the model for the classification task. This task is also dependent on the type or source of data and the results of the classifiers varies to a great extent as a result of pre – processing. To prepare the data various steps are performed like removal of special characters, stop word removal and tokenization. Special characters refer to the characters like comma, full stop etc. which are removed from the raw data. The next step consists of conversion of the data strings into tokens. The tokens are individual words that do not have any non – alphabetic character in between. Along with alphabetic tokens numeric tokens are also retained. These tokens form the set of feature space for the classifier models. From this feature set, the most commonly occurring words, also known as stop words are removed since these don't contribute significantly to the feature set. Apart from these basic steps, twitter data need some additional pre – processing due to the nature of tweets. Since tweets that have been scraped are public live tweets therefore, each consists of a hyperlink that opens the tweet in the Twitter App. These links are removed from the raw data. Tweets also contain the twitter handle, i.e. twitter user name of the person sending the tweet. Since our work is based on the text features, therefore the twitter handle (starting with @ followed by the user name) is removed which is not significant for the text classification work.

B. Dataset

A supervised learning classifiers are used to carry the classification tasks. Therefore, training as well as testing data is required for effective classification. The spam classification is carried on the short social media text messages that are limited in length. We have considered two mediums of social text: the mobile messages in the form of SMS text and text from micro blogging site, Twitter. The data for the SMS message is taken from the UCI Repository and the Twitter text data is scrapped from the public live tweets. These two datasets are described below:

SMS Spam Corpora

SMS Spam Collection dataset is taken from UCI repository which was composed in 2012. It consists of 5574 mobile SMS out of which 747 are spam, and 4827 are ham. These text messages were collected from various sources like 425 spam messages were taken from the UK website: Grumbletext, 3,375 ham SMS from NUS SMS Corpus (NSC), 450 ham messages from Caroline Tag's PhD Thesis, and rest 1002 messages from SMS Spam Corpus v.0.1 [37]. The dataset is in the form of text file where every line consists of a label followed by the message. The distribution of text SMS as spam and ham is shown in Table 1. An example of SMS spam is shown Figure 5:



+918471025341

2015/11/12 Thu

YOUR MOBILE NO HAVE WON 2.6CRORE IN INDIA/UK MICROMAX 2015 AWARD TO CLAIM YOUR PRICE SEND YOUR NAME,ADDRES,PHONE NO,AGE,OCCP

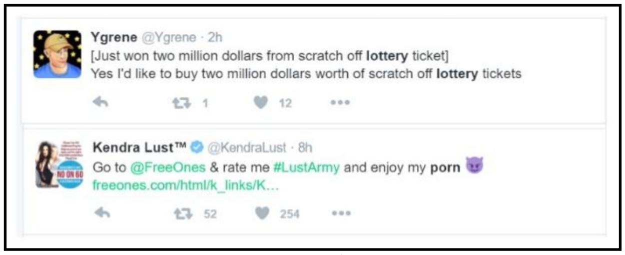
TO.micromaxpriceclaim@hotmail.com

7:33 PM

Figure 5. Example of SMS Spam

Twitter Corpora

This dataset has been created by scrapping the public live tweets from the micro blogging site Twitter using Twitter API. While scrapping the tweets, the keywords were provided that might help us to retrieve the desired type of tweets falling into one of the categories of spam or ham. Some examples of the keywords or the lexicons are "porn", "lottery", "school", "video" etc. The full list is provided in Appendix B. These tweets have been manually classified as ham or spam. Some examples of Twitter spam are shown in Figure 6:



Ygrene @Ygrene · 2h
[Just won two million dollars from scratch off **lottery** ticket]
Yes I'd like to buy two million dollars worth of scratch off **lottery** tickets

1 12

Kendra Lust™ @KendraLust · 8h
Go to @FreeOnes & rate me #LustArmy and enjoy my **porn**
freeones.com/html/k_links/K...

52 254

Figure 6. Examples of Twitter Spam

The no. of tweets class wise are shown in the Table 2.2

Table 1. Details of Spam Corpus

Dataset Description			
Dataset	Total Instances	No. of Hams	No. of Spam
SMS Spam	5574	4827	747
Twitter	5096	4231	865

The dictionary of words is created for both the datasets where the SMS dataset has 85,477 words with 8,277 unique words. The Twitter dataset has 97,831 words with 14538 unique words.

C. Evaluation Measures

For the evaluation of results, metrics like Precision, Recall, Accuracy and F1 score are calculated. The calculation of these matrices is based on the confusion matrix shown in Table 2. For a binary classification problem the matrix has 2 rows and 2 columns. Across the top, the labels represent the actual class labels and down the side, the predicted class labels are shown. Each cell in the matrix shows the number of predictions by the classifier of the category of that cell. Various labels of matrix are defined as:

TP – Positive labels predicted as positive

TN – Negative labels predicted as negative

FP – Negative labels wrongly identified as positive

FN – Positive labels wrongly predicted as negative

Table 2. Confusion Matrix

Type	Ham	Spam
Ham	True Positive (TP)	False Positive (FP)
Spam	False Negative (FN)	True Negative (TN)

The various evaluation metrics are calculated as below:

1. *Accuracy* is the number of correct predictions made in either of the class divided by the total number of predictions made. It is then multiplied by 100 for getting the percentage. It is calculated as shown in Equation 3:

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + FP + FN + TN}} \qquad \textbf{(3)}$$

2. *Precision* is the number of True Positives divided by the total of True Positives and False Positives. Thus, Precision is the measure of a classifiers exactness. A low precision might indicate a large number of False Positives. It is calculated as shown in Equation 4:

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \qquad \textbf{(4)}$$

3. *Recall* is the number of True Positives divided by the total number of True Positives and False Negatives. A recall can be thought of as a measure of a classifiers completeness. A low recall indicates high False Negatives. It is calculated as shown in Equation 5:

$$\boldsymbol{Recall = \frac{TP}{TP + FN}} \qquad \textbf{(5)}$$

4. *F1 Score* is one of the best measure of classifier's accuracy. While calculating F1 scores, precision and recall both are considered as it is the weighted average of both. It has the value between 0 and 1 while 0 being the worst case and 1 being the best case. It is calculated as shown in Equation 6:

$$\boldsymbol{F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}} \qquad \textbf{(6)}$$

D. Results and Discussion

The experiments for the classification of Spam on traditional classifiers are tested on two corpuses: SMS spam dataset and Twitter dataset. It is implemented in python 2.7 using scikit-learn library. In the experiments, bag – of – words model is used for text representations with two different weighing schemes for feature extraction: Term Frequency (TF) and Term Frequency – Inverse Document Frequency (TF – IDF) to extract feature vectors.

The application of the Naïve Bayes algorithm for the spam classification is carried out with the variable feature size. 10 fold cross validation is used with 80% data used for training while 20% used for testing which resulted in accuracy of 97.65% in the SMS spam dataset and 91.14% in the case of Twitter dataset.

The results of the application of support vector machine on both the datasets with various kernel functions are shown in Table 3. The performance of a linear function is better as compared to others.

Table 3. Accuracy of SVM with different kernel functions

Kernel Function	Accuracy (SMS)	Accuracy (Twitter)
Linear	97.45	93.14
Polynomial Degree = 2	87.71	82.45
Polynomial Degree = 3	85.65	83.24

Table 4 shows classification of spams based on different value of k for both the sub tasks. The performance shows that with the increase in the number of neighbors, the accuracy of the classifier generally drops.

Table 4. Accuracy of KNN classifier with different value of k

K	Accuracy (SMS)	Accuracy (Twitter)
2	90.40	91.96
5	87.67	89.60
10	89.87	87.80
20	85.92	88.79

The implementation of random forest is done using 100 estimators. Though the complexity of the model has increased, but there is an increase in performance in terms of accuracy as compared to the other classification models.

Table 5. Classifier Results SMS Spam Corpus with weighing scheme: TF

Classifier	Precision	Recall	Accuracy	F1
KNN	88.60	86.82	86.82	82.52
NB	97.74	97.76	97.76	97.74
RF	96.89	96.77	96.77	96.62
ANN	97.38	97.40	97.40	97.39
SVM	97.18	97.22	97.22	97.18

Table 6. Classifier Results SMS Spam Corpus with weighing scheme: TF – IDF

Classifier	Precision	Recall	Accuracy	F1
KNN	91.37	90.40	90.40	88.13
NB	97.64	97.67	97.67	97.65
RF	97.88	97.85	97.85	97.77
ANN	97.41	97.40	97.40	97.40
SVM	97.45	97.49	97.49	97.44

Table 7. Classifier results Twitter Corpus with weighing scheme: Term Frequency

Classifier	Precision	Recall	Accuracy	F1
KNN	89.56	89.41	89.41	89.48
NB	91.21	91.57	91.57	91.14
RF	91.78	91.27	91.27	90.18
ANN	91.36	91.67	91.47	91.39
SVM	91.79	92.06	92.06	91.60

Table 8. Classifier results Twitter Corpus with weighing scheme: TF – IDF

Classifier	Precision	Recall	Accuracy	F1
KNN	91.61	91.96	91.96	91.38
NB	91.69	92.06	92.06	91.74
RF	93.25	93.43	93.43	93.04
ANN	91.80	91.18	91.18	91.41
SVM	92.91	93.14	93.14	92.97

These classifiers are the benchmark classifiers for comparing the results of the spam detection and they work on the features extracted before – hand to develop a classification model that could be used with the unlabeled data. The classification results for various classifiers with 2 different weighing schemes are summarized in Table 5 to 8. The experimental results show that the Random Forest (97.88 and 93.43) performs best for the spam classification task using *TF-IDF* weighing scheme, among the traditional classifiers with an accuracy of 97.88% for SMS spam dataset and 93.43% for Twitter dataset.

IV. Conclusion

Experiments were performed on various classifiers that are used as benchmark models for spam detection. These traditional classifiers performed well in terms of accuracy in classification of spams in both the dataset. The features are extracted by the feature extraction method: bag – of – words, using two different weighing schemes: *TF* and *IDF*. In both the datasets, it was found that the results of the *TF-IDF* are better and prevents overfitting of data. Overall, Random Forest performs the best with 100 estimators with an accuracy of 97.77% in case of SMS spam corpus and an accuracy of 93.04% in case of Twitter corpus. The

The SMS text message and Tweets being limited in size, have a limited number of features and also people use shorthand notations, unstructured English language with lot of slangs. Therefore, the classifier training is difficult and it is best done with large amount data. The future work includes the use of pre – trained word vectors such as Google's word2vec which adds semantic meaning to the word vectors for classification tasks. The feature extraction is handcrafted, therefore a method is sought after where the model learns the features, instead of depending on features extracted by the human experts.

REFERENCES

[1] L. F. Cranor, & B. A. LaMacchia, "*Spam!*", Communications of the ACM, Vol 41, Issue 8, pp.74-83. 1998.

[2] G. Jain, & M. Sharma, "*Social Media: A Review*", In Information Systems Design and Intelligent Applications, Springer India, pp. 387-395. 2016

[3] M. Nelson, "*Spam Control: Problems and Opportunities*", Ferris Research, India, pp.23-82, 2003.

[4] G.V. Cormack, J.M.G. Hidalgo, E.P. Sánz, "*Feature engineering for mobile (SMS) spam filtering*", In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, US, pp. 871-872, 2007.

[5] C. K. Grier, Thomas, V. Paxson, M. Zhang, *"@ spam: the underground on 140 characters or less*", In Proceedings of the 17th ACM conference on Computer and communications security, Chicago, pp.27-37, 2010.

[6] H. Stern, "*A Survey of Modern Spam Tools*", In Proceedings of the 5th Conference on Email and Anti-spam, CA, pp.1-10, 2008.

[7] A. Cournane, R. Hunt, "*An analysis of the tools used for the generation and prevention of spam*", Computers & Security, Vol.23, Issue.2, pp.154-166, 2004.

[8] T. Kurt, C. Grier, J. Ma, V. Paxson, D. Song, "*Design and evaluation of a real-time url spam filtering service*", In 2011 IEEE Symposium on Security and Privacy, USA, pp.447-462, 2011.

[9] J. Kim, K. Chung, K. Choi, "*Spam filtering with dynamically updated URL statistics*", IEEE Security & Privacy, Vol.4, Issue.5, pp.33-39, 2007.

[10] J.R. Levine, "*Experiences with Greylisting*", In Proceedings of 2nd Conference Email and Anti-Spam (CEAS 05), NY, pp1-2, 2005

[11] B. Agarwal, N. Mittal, "*Prominent feature extraction for review analysis: an empirical study*", Journal of Experimental & Theoretical Artificial Intelligence, Vol.28, Issue.3, pp.485-498, 2016.

[12] B. Agarwal, N. Mittal, "*Sentiment analysis using conceptnet ontology and context information*", In Prominent Feature Extraction for Sentiment Analysis (Springer), US, pp.63-75, 2016.

[13] L. Zhang, J. Zhu, T. Yao, "*An evaluation of statistical spam filtering techniques*", ACM Transactions on Asian Language Information Processing (TALIP), Vol.3, Issue.4, pp.243-269, 2004.

[14] I. Rish, "*An empirical study of the naive Bayes classifier*", In IJCAI 2001 workshop on empirical methods in artificial intelligence, Vol.3, Issue.22, pp.41-46, 2001.

[15] F. Sebastiani, *"Machine learning in automated text categorization*", ACM computing surveys (CSUR), Vol.34, Issue.1, pp.1-47, 2002

[16] Z. Yang, X. Nie, W. Xu, J. Guo, "*An approach to spam detection by naive Bayes ensemble based on decision induction*", In Sixth International Conference on Intelligent Systems Design and Applications, China, pp.861-866, 2006

[17] C. Kim, K. B. Hwang, "*Naive Bayes classifier learning with feature selection for spam detection in social bookmarking*", In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/ PKDD), US, pp.32, 2008.

[18] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, C. D. Spyropoulos, "*An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages*", In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Greece, pp.160-167, 2000.

[19] T. A. Almeida, A. Yamakami, J. Almeida, "*Evaluation of approaches for dimensionality reduction applied with naive bayes anti-spam filters*", International Conference on Machine Learning and Applications, Miami, pp.517-522, 2009.

[20] C. Cortes, & V. Vapnik. "*Support-vector networks*", Machine learning, Vol.20, Issue.3, pp.273-297, 1995

[21] M. Mccord, M. Chuah, "*Spam detection on twitter using traditional classifiers*", In International Conference on Autonomic and Trusted Computing, Heidelberg, pp.175-186, 2011.

[22] P. Kolari, T. Finin,, A. Joshi, March, "*SVMs for the Blogosphere: Blog Identification and Splog Detection*", In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, Baltimore, pp.92-99, 2006.

[23] H.B. Wang, Y. Yu, Z. Liu, "*SVM classifier incorporating feature selection using GA for spam detection*", In International Conference on Embedded and Ubiquitous Computing, Japan, pp.1147-1154, 2005

[24] C. Y. Tseng, M. S. Chen, "*Incremental SVM model for spam detection on dynamic email social networks*", In Int. Conf. on Computational Science and Engineering, Vancouver, pp.128-135, 2009.

[25] M. Healy, S. J. Delany, A. Zamolotskikh, "*An assessment of case base reasoning for short text message classification*", In N. Creaney (Ed.), Proceedings of 16th Irish Conference on Artificial Intelligence and Cognitive Science, Castlebar, pp.257-266, 2005.

[26] A. Harisinghaney, A. Dixit, S. Gupta, A. Arora, "*Text and image based spam email classification using KNN Naïve Bayes and Reverse DBSCAN algorithm*", In Optimization Reliabilty and Information Technology (ICROIT), India, pp.153-155, 2014

[27] T.P. Ho, H.S. Kang, S.R. Kim, "*Graph-based KNN Algorithm for Spam SMS Detection*", Journal of Universal Computer Science, Vol.19, Issue.16, pp.2404-2419, 2013.

[28] F. Barigou, B. Beldjilali, B. Atmani, "*Using cellular automata for improving knn based spam filtering*", Internationa Arab Journal Information Technology, Vol.11, Issue.4, pp.345-353, 2014.

[29] A.T. Sabri, A. H. Mohammads, B. Al-Shargabi, M. A. Hamdeh, "*Developing new continuous learning approach for spam detection using artificial neural network (CLA_ANN)*", European Journal of Scientific Research, Vol.42, Issue.3, pp.525-535, 2011.

[30] MR. Nagpure, SS. Mesakar, SR. Raut and Vanita P.Lonkar, "*Image Retrieval System with Interactive Genetic Algorithm Using Distance*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.12, pp.109-113, 2014.

[31] D. DeBarr, & H. Wechsler, "*Spam detection using clustering, random forests, and active learning*", In Sixth Conference on Email and Anti-Spam. Mountain View, California, pp.1-6, 2009.

[32] A. Karami, L. Zhou, "*Improving static SMS spam detection by using new content-based features*", In 20th Americas Conference on Information systems (AMCIS), Savannah, pp.1-9, 2014.

[33] A. Garg, N. Batra, I. Taneja, A. Bhatnagar, A. Yadav, S. Kumar, "*Cluster Formation based Comparison of Genetic Algorithm and Particle swarm Optimization Algorithm in Wireless Sensor Network*", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.2, pp.14-20, 2017.

[34] Y. Zhang, S. Wang, P. Phillips G. Ji, "*Binary PSO with mutation operator for feature selection using decision tree applied to spam detection*", Knowledge-Based Systems, Vol.64, Issue.3, pp.22-31, 2014.

[35] DJ. Hand, K. Yu, "*Idiot's Bayes—not so stupid after all?*", International statistical review, Vol.69, Issue.3, pp.385-398, 2001

[36] LBreiman, "*Random forests*", Machine learning, Vol.45, Issue.1, pp.5-32, 2001.

[37] M. Lichman, "*UCI Machine Learning Repository*", School of Information and Computer Science University of California, California, pp.1-143, 2013.

Authors Profile

Gauri Jain pursued Masters of Technology in Computer Science from ITM (MD University), Gurgaon and is currently pursuing her Ph.D. in Computer Science and Engineering from the Banasthali University, India. She has published 4 papers in international journals and conferences. She has 2 years of experience as a software developer and more than 5 years of teaching experience. Her main area of interest includes Deep Learning, Machine Learning, Spam Detection. and Text Mining.

Dr. Manisha is a former Associate Professor at Banasthli University, Banasthal and have around 18 years of experience. She has done her Ph. D. in Computer Science from Banasthali University and published more than 30 research papers in reputed national and international journals. Her main research work focuses on Artificial Intelligence, Data Mining and Information Retrieval.

Dr. Basant Agarwal is an associate professor at SKIT Jaipur India. He has a PhD in Computer Science and Engineering from Malaviya National Institute of Technology, Jaipur, India. His research interest is in natural language processing, machine learning, deep learning, sentiment analysis and opinion mining. He has published several research papers in international conferences and journals of repute.