

# Machine learning based Indian spam recognition

Amartya Chakraborty

Technical Assistant, Department of CSE  
Techno India University  
Kolkata, West Bengal  
amartya3@gmail.com

Sangita Karmakar

M.Tech. Student, Department of CSE  
Techno India University  
Kolkata, West Bengal  
sangitakarmakar1995@gmail.com

Suwendu Chattaraj

Assistant Professor, Department of CSE  
Techno India University  
Kolkata, West Bengal  
suwendu.chattaraj@rediffmail.com

**Abstract**—The Short Message Service or SMS has prevailed as a very popular communication channel in mobile phone users since its early advent. However, in this day and age of web-based instant messaging applications, this service has indeed lost its former dependence. Instead, now SMS has turned into the forte of spammers. In this work, a easily available, popular SMS data set has been used, which is modified by adding both regional spam and ham texts that are typed in english. Thereafter the new set of data is processed, features are extracted and then classified by using three widely used classification algorithms, to provide a enriched recognition system that is more suited to identifying SVM spams in the Indian context. Experimental results show that SVM performs most robustly among the classifiers used in our work, as determined by a Monte Carlo approach.

**Index Terms**—SMS spam; Spam filtering; Natural language processing; Supervised learning; Text classification.

## I. INTRODUCTION

Humans have practiced different means of communication over the years, such that they can be adapted even at a distance. One such modern, efficient mode of electronic communication is the Short Message Service (SMS), which dates back to the year 1992 [1]. A mobile device user can compose a secure SMS with a maximum of 160 alphanumeric characters [2] to convey a short message to the recipient. Such mode of communication is especially useful in cases where it is infeasible to attend to a call, or when it is required to convey a small piece of urgent information. However, over the years, this same service, has become a *marketing tool*, and is now extensively used in *direct marketing*, as discussed in [3]. Reports show that in 2014 alone, the SMS based marketing business was worth over a \$100 billion, and was expected to generate revenues of over \$1.7 trillion! [4]. The direct repercussion of this development is evident from the current scenario in India. *Direct marketing* has enabled the business organizations to pre-identify their potential clients, and approach them directly via electronic messages with offers that they might find attractive. Another recent study [5] has shown that the average Indian smart-phone user receives between four to seven unwanted marketing messages every day. This is despite the fact that there are regulatory norms in place administered by Telecom Regulatory Authority of India (TRAI). Most of the time, the survey found that the telecom company itself is the biggest sender of unwanted marketing messages via SMS. This sort of a situation necessitates an understanding of what *unwanted messages* denotes, which brings us to the discussion about spam messages.

Fig. 1. Screen-shot of a sample SPAM SMS

The word spam essentially denotes *unsolicited* or *unwanted* electronic messages that is sent primarily by marketing companies to reach potential clients. It is a curse of today's technological advancement, that in most of the cases the smart-phone users hardly realize that they are signing up for spam messages whenever they avail a service from a company of their choice. For example, a persons bank may keep sending daily SMS messages about newer offers or different account interest rates, which may be undesirable. The same thing happens for their online shopping websites, their telecommunication providers, etc. In other cases, SPAM can be completely malicious or fraudulent messages aimed at extracting vital information about the target's financial details, as shown in Figure 1. On the other hand, HAM messages are the *desired* electronic communications received by a smart-phone user. These could be messages from their acquaintances, own account related updates from their banker, or travel ticket information. So, any non-spam message or wanted or useful message is referred to as a HAM message. In the following section, we take a look at the works of different researchers over the years, in the context of spam identification.

## II. LITERATURE SURVEY AND MOTIVATION

In this section, the work of some different researchers in the domain of SMS spam filtering have been discussed in a chronological order. These works deal with different problems, such as identification of spam and effective classification mechanisms, use of novel features, deployable smart-phone based solutions etc. Unlike the popular approach of spam identification at the user end on hand held smart devices, Dixit *et al.* [6] proposed a filtering mechanism at the Short Message Service Center (SMSC) itself, whereas, most of the other works follow a different approach. The main source of data for our current work is the text corpus aggregated and used by Almeida *et al.* [7]. This data set contains non-encoded spam and ham messages collected mainly in UK and Singapore. From their evaluation, it was seen that SVM with linear kernel outperforms all other classification approaches for their text corpus. In the same year, Yadav *et al.* [8] used an India-centric corpus of ham and spam messages, collected

via crowd sourcing in the IITD campus. A novelty present in this work was that the ham messages consisted of *regional words typed in English*. They have also used SVM along with Bayesian learning to evaluate the classification accuracy of their proposed system. The results have shown that Bayesian learning performs as good as SVM in SMS classification. The same was observed by Mathew *et al.* [9]. In general, the processing and classification of huge volumes of text data is a computationally heavy task, and is not really feasible to perform on an isolated smart-phone. The work by Taufiq *et al.* [10] addressed this issue and provided a probabilistic Naive Bayes classifier for screening and identification of spam at the user end. A similar work by Uysal *et al.* [11] devised a spam detection framework that could be deployed in mobile phones, and evaluated it on a collection of SMS texts. Xu *et al.* [12] proposed an interesting non-content based spam identification system which was privacy preserving, as it did not use the SMS content at all. On the other hand, using text based features, Almeida *et al.* [13] worked with the classification of another huge collection of spam and ham messages, and determined that SVM outperforms most of the other classification algorithms here too. A similar finding by Shirani-Mehr *et al.* [14] strengthened the former observation about the performance of SVM based linear kernel with the use of 10-fold cross validation technique. Concurrently, researchers have also explored the different novel, content based features, by which spam could be better identified, like the work by Karami *et al.* [15]. Narayan *et al.* [16] determined the effectiveness of different smart-phone based spam-detecting applications developed throughout the years and proposed a two level classifier for more accurate performance. The work by Agarwal *et al.* [17] has extended the corpus provided and used originally by Almeida *et al.* [7], and also added the Indian context to it. For evaluation purposes, both classification (SVM, Naive Bayes, Random Forest) and clustering (k-means) has been used. The results show that MNB classifier performs almost as good as SVM, with a lesser computation time. In recent times, the classification of text messages has been proposed using other methods like deep learning and convolutional neural networks [18].

Motivated by the above observations, the authors have extended the SMS message corpus provided by Almeida *et al.* [7], by adding regional language based spam and ham messages to it. Initially, this data set has been duly processed and then three classifiers, namely Support Vector Machine (SVM), k-Nearest Neighbours (kNN) and Decision Tree (DT) have been chosen. Finally the authors have attempted to determine the robustness of the classifiers with a monte carlo approach using k-fold cross validation method for a considerably large value of k.

### III. DESCRIPTION OF THE TEXT CORPUS

For the initial experimentation, the original data set provided in [19] and studied comprehensively by [7] has been used. This corpus contains a labelled collection of both spam and ham messages gathered from other SMS corpora as described

in [19]. Then the authors have further extended this data set by adding the context of Indian spam. For this purpose, a set of both spam and ham messages has been collected from the faculties of Techno India University, West Bengal. An interesting factor that is introduced in this extended set of messages, is a collection of regional texts typed in English. Such messages mostly consist of Hindi words typed in English font. An example of such a spam message is as given below in Figure 2

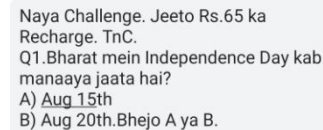


Fig. 2. Screenshot of regional spam text typed in English

### IV. DATA PROCESSING

The raw SMS data consists of labelled “ham” and “spam” messages kept in two columns, and as described in Section III. For processing any form of textual data, different methods are used, for instance, a piece of text needs to be *cleaned*, or *normalized*, such that any forms of noise in the data is done away with, semantic grouping is possible, and data can be represented in a simpler manner without losing the underlying features. Some of the methods used in this work are:

- **Case Normalization (CN)** ensures that all the words in a text belong to the same case, i.e. either upper case or lower case. For instance, “Hello” can be normalized to “HELLO” or “hello” using CN.
- **Stop-Word Removal (SWR):** The *useless, common* words or data in a text are called stop words. Some examples of such stop words are “a”, “an”, “in”, “because”, “the”, etc.
- **Stemming or Lemmatization:** this refers to the deduction of the *word of origin* or the *root form* of a particular word. For example, “am”, “are”, “is” are *inflected words*, or appropriately modified forms of the word “be”.
- **Tokenization:** This is the splitting of textual data in to a small chunk of words, such that they find representation as individual *tokens*. This forms the basis of lexical analysis and is a very commonly used tool.

All the aforementioned methods have been used for processing the text in this work, except stemming, as it is ineffective on colloquial english. Also, in the implemented system, tokenization is carried out inherently in the feature extraction procedure, as discussed in the next section.

### V. FEATURE EXTRACTION

After the text was processed, the different features have been extracted from it, so as to make it classifiable in a computationally feasible manner using **Vectorization**. This method converts a processed text into a matrix of numerical values that are used to train a classifier. There are different types of vectorizers that can be used for the conversion of text to numerical values, and in this work the authors have employed the **TF-IDF vectorizer** (Term Frequency Inverse

Document Frequency) to formulate a vector representation of our text messages. This choice is based on its general efficiency in vectorizing text documents [20]. The TF-IDF attempts to determine the importance of every word in the corpus of the data set, as discussed below.

**Term-Frequency (TF):** Every word in a text document is called a *term*, the frequency of occurrence of each term is known as its *Term Frequency*. Longer sentences may contain the same word more often than a shorter sentence. Mathematically, the Term Frequency (TF) is defined as:

$$TF = \frac{\text{No. of times the word appears in the corpus}}{\text{Total no. of words in the corpus}} \quad (1)$$

**Inverse Document Frequency (IDF):** the weightage or importance of a word in a document is determined using the following logic - the more the frequency of a word in the document, the lesser is its importance. The importance of a word  $w$  in a document can be expressed as:

$$\text{Importance}_w \propto \frac{1}{\text{Frequency}_w} \quad (2)$$

Consequently, the Inverse Document Frequency is represented as:

$$IDF = \log_{10} \frac{\text{Total no. of documents}}{\text{No. of documents in which the word appears}} \quad (3)$$

**Calculation of TF-IDF:** The last step is to multiply the resultant TF and IDF values of each word in the document. This gives the weightage of the TF of every word in a document against its IDF value. The least common words, as a result, have more weightage and vice-versa. Mathematically the TF-IDF score for a word  $w$  is represented as:

$$TFIDF_w = TF_w * IDF_w \quad (4)$$

This TF-IDF matrix is then used as a feature set for further experimentation using classification algorithms, as discussed in the next section.

## VI. EXPERIMENTAL RESULTS WITH ORIGINAL CORPUS

In this section, the results of the different experiments performed on the original, unmodified data have been put up. The processing of the text has been done in two parts, in the first case, CN has been used in isolation and in the other, both CN and SWR have been combined prior to vectorization of the resultant text. In each case, for classification purposes, 3 different classifiers have been used, namely SVM, kNN and DT. The authors have used constant, standard values for parameters throughout the evaluation, and used constant stratified testing to ensure proper representation of the labels. Also, a k-fold cross validation method has been used, in order to eliminate any dependence on the manner of splitting of data, and avoid holdout.

### A. Results with the original data set using only CN

The first part of the experiment was conducted using only CN on the data, followed by TF-IDF vectorization. The vectorized data has then been classified and evaluated using 10-fold cross validation. Table I shows the accuracy scores for every fold of computation recorded for the classifiers individually. The distribution of these scores have also been illustrated in the Figure 3.

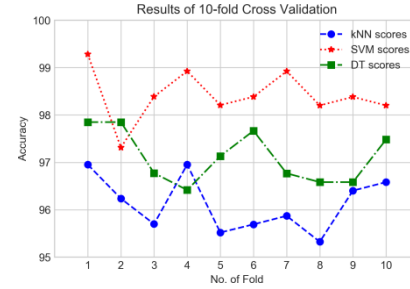


Fig. 3. Comparison of classifier performance with 3 classifiers, using original corpus processed by only CN

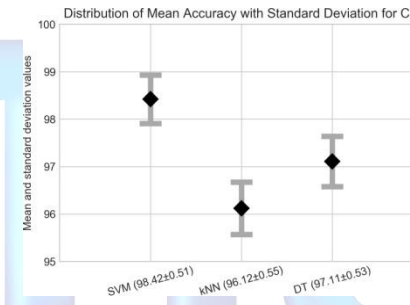


Fig. 4. Distribution of Mean Accuracy with Standard Deviation for 3 classifiers, using original corpus processed by only CN

In this case, it is found that SVM performs far better than both DT and kNN in classification of the vectorized data set, with a maximum accuracy of 99.28% and a mean accuracy of 98.42%. This was followed by DT classifier with a maximum accuracy of 97.84%, and kNN performs the worst among all the three. The distribution of the Mean Accuracy and Standard Deviation for the 3 different classifiers has been shown in the Figure 4.

### B. Results with the original data set using CN and SWR

In this part of the experiment, the original data set has been processed using both CN and SWR. After this, TF-IDF vectorization has been used as earlier, and the vectorized data has been classified and evaluated using 10-fold cross validation. Table II contains the accuracy scores of the respective classifiers in every fold of computation. This has also been illustrated in Figure 5. The distribution of mean accuracy and standard deviations for each classifier have been shown in the Figure 6. As in the previous case, it is seen that a maximum accuracy is achieved by SVM (with linear kernel) with 99.64%, followed closely by DT, and kNN falls behind both in terms of classification accuracy.

Classifier	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	Mean	Time(s)
SVM	99.28	97.31	98.38	98.92	98.21	98.38	98.92	98.20	98.38	98.20	98.42	15.79
kNN	96.95	96.23	95.69	96.95	95.51	95.69	95.87	95.32	96.40	96.58	96.12	3.19
DT	97.84	97.84	96.77	96.41	97.13	97.66	96.76	96.58	96.58	97.48	97.11	8.29

TABLE I  
RESULTS OF THE 10-FOLD CROSS VALIDATION ON THE ORIGINAL DATASET WITH ONLY CN

Classifier	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	Mean	Time(s)
SVM	99.64	97.67	98.74	98.74	97.49	98.38	98.74	98.02	98.56	98.38	98.43	12.58
KNN	95.34	95.34	93.18	94.98	94.26	94.25	93.72	94.24	95.50	94.96	94.58	2.76
DT	98.56	97.13	96.77	97.84	97.67	97.30	96.94	95.50	96.40	97.84	97.19	5.33

TABLE II  
RESULTS OF THE 10-FOLD CROSS VALIDATION ON THE ORIGINAL DATA SET WITH CN AND SWR

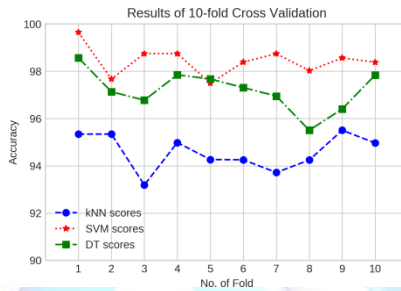


Fig. 5. Comparison of classifier performance with 3 classifiers, using original corpus processed by CN and SWR

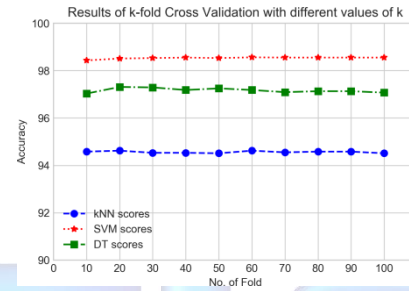


Fig. 7. Distribution of mean accuracy for large values of k, using 3 classifiers on data processed by CN and SWR

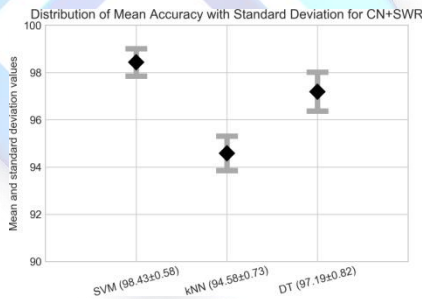


Fig. 6. Distribution of mean accuracy with standard deviation for 3 classifiers, using original corpus processed by CN and SWR

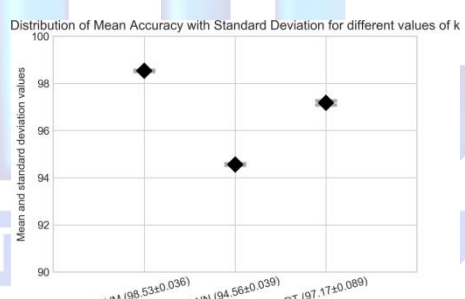


Fig. 8. Distribution of mean of mean accuracy and standard deviation for 3 classifiers with large values of k upto 100

### C. Results with large values of K in Cross Validation

The previous experiments have shown a very good performance during testing by k-fold cross validation (with  $k=10$ ). However, as discussed earlier, the value of  $k=10$  is indeed a standard value used for cross-validation. The process of cross-validation using k number of folds can be repeated for any value of  $k \leq n$ , where n is the number of samples in the data set. A larger value of k gives more randomness of samples to the classifier, and as such it's performance can be judged better. Here, the CN + SWR processed data is evaluated using the same classifiers. However, instead of 10-fold cross validation only, the authors have evaluated the data using a value of k between 10 to 100, at intervals of 10 folds.

The corresponding mean accuracy scores for every case has been shown in Table III, along with the mean and standard deviation of all the cases taken together up to 100 folds. Figure

7 shows the distribution of the mean accuracy values for all the k fold evaluations, while the statistical distribution of the mean accuracies for  $k=100$  folds, and the standard deviation for each classifier, have been illustrated in Figure 8.

**Observations:** From the Tables I and II, it is seen that with the consecutive use of two data processing methods CN and SWR, the mean accuracy has increased fractionally for SVM and DT classifiers. Thus the change in the structure of the texts by using SWR along with CN has resulted in better classification. In the case of the k-NN classifier only, a fall in the accuracy can be noticed. This may be due to the standard value of k chosen in this work. Also, from the Table III, it is evident that the mean of mean accuracies is maximum in the case of SVM, and least in the case of kNN. This is in corroboration with the findings of 10-fold cross validation discussed in the previous sections. Again in comparison to the mean of the accuracies

Classifier	k=10	k=20	k=30	k=40	k=50	k=60	k=70	k=80	k=90	k=100	Mean	Std. Dev.
SVM	98.43	98.51	98.53	98.55	98.53	98.56	98.55	98.55	98.55	98.55	98.53	0.036
KNN	94.58	94.62	94.53	94.53	94.51	94.62	94.55	94.58	94.58	94.51	94.56	0.039
DT	97.03	97.31	97.29	97.18	97.25	97.18	97.09	97.13	97.13	97.07	97.17	0.089

TABLE III  
RESULTS OF THE K FOLD CROSS VALIDATION ON THE ORIGINAL DATA SET WITH CN + SWR AND  $10 \leq k \leq 100$ ,  $k=k+10$

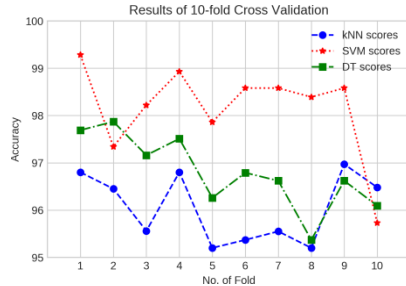


Fig. 9. Evaluation of modified corpus processed by CN

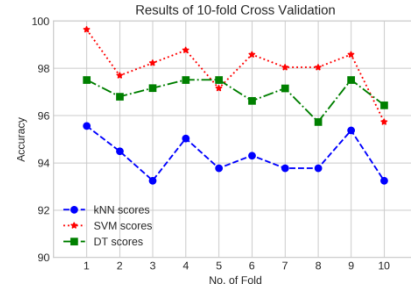


Fig. 11. Evaluation of modified corpus processed by CN and SWR

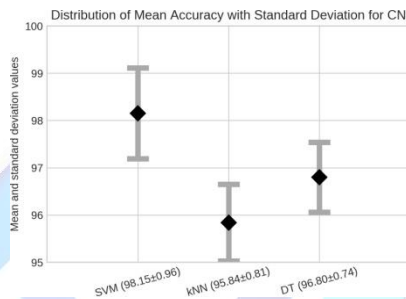


Fig. 10. Distribution of mean accuracy with standard deviation for 3 classifiers on modified corpus processed by CN

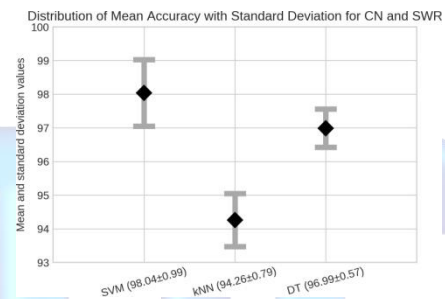


Fig. 12. Distribution of mean accuracy with standard deviation for 3 classifiers on modified corpus processed by CN and SWR

of 10-fold cross validation using CN and SWR, an increase is seen only in the case of SVM, though fractional. So, can be determined that SVM performs consistently in terms of classification in this work, and also in general for a given nlp based problem [21].

## VII. EXPERIMENTAL RESULTS WITH MODIFIED CORPUS

This is the second part of the experiment where the modified SMS corpus has been used, as described in Section III. As in the previous experiment, the SVM, KNN and DT classifiers with similar constant parameter values have been used all throughout. For proper comparable classification, k-fold cross validation has also been used as discussed below.

### A. Results with modified data set using only CN

Here the new corpus has been processed using *case normalization* (CN), followed by vectorization, and classification. The performance of the classifiers has been evaluated with 10 fold cross validation, the results of which have been illustrated in Table IV and Figure 9.

From the results, it is noticed that SVM again performs the best among the three classifiers, with a maximum accuracy of 99.28%, and kNN gives the least accuracy of with a maximum of 96.8%. DT classifier comes second, but outperforms SVM once in the 10 fold evaluation process. Similarly, kNN also

outperforms DT classifier in one particular fold of evaluation. The corresponding mean and standard deviation values for all the 3 classifiers have been put up in Figure 10.

### B. Results with modified data set using both CN and SWR

In this case, both CN and SWR have been used on the modified corpus, followed by classification using the three classifiers. The corresponding accuracy scores have been illustrated in Table V.

The Figures 11 and 12 illustrate the accuracy for each fold of evaluation, and the mean and standard deviation values for each classifier in this case. Here, unlike the previous experiment discussed in Section VI, it is found that only DT classifier improves its performance fractionally due to the use of CN and SWR. This may be due to the change in the branching structure, and consequently in decision making nodes.

### C. Results with large values of K in Cross Validation

This is the final part of the experiment, where the mean of mean accuracies for all the three classifiers have been calculated using a fold interval of 10. The modified text corpus is processed by both CN and SWR, and then classified, same as the experiment conducted in Section VI-C. This is aimed at gaining a better clarity about the performance of the

Classifier	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	Mean	Time(s)
SVM	99.28	97.34	98.22	98.93	97.86	98.58	98.58	98.39	98.58	95.73	98.15	23.49
kNN	96.80	96.45	95.56	96.80	95.20	95.37	95.55	95.20	96.97	96.48	95.84	4.64
DT	97.69	97.87	97.16	97.51	96.26	96.79	96.62	95.37	96.62	96.09	96.80	10.39

TABLE IV  
RESULTS OF THE 10-FOLD CROSS VALIDATION ON THE MODIFIED DATA SET WITH ONLY CN

Classifier	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	Mean	Time(s)
SVM	99.64	97.69	98.22	98.76	97.15	98.58	98.04	98.04	98.58	95.73	98.04	19.46
KNN	95.56	94.49	93.25	95.03	93.77	94.31	93.77	93.77	95.37	93.24	94.26	3.32
DT	97.51	96.80	97.16	97.51	97.51	96.62	97.15	95.73	97.51	96.44	96.99	8.50

TABLE V  
RESULTS OF THE 10-FOLD CROSS VALIDATION ON THE MODIFIED DATA SET WITH CN AND SWR

Classifier	k=10	k=20	k=30	k=40	k=50	k=60	k=70	k=80	k=90	k=100	Mean	Std. Dev.
SVM	98.04	98.20	98.22	98.20	98.22	98.26	98.27	98.31	98.35	98.36	98.24	0.088
KNN	94.26	94.39	94.39	94.34	94.38	94.39	94.34	94.38	94.35	94.35	94.36	0.038
DT	96.89	97.21	97.07	96.99	96.96	96.83	96.94	96.83	96.99	96.88	96.96	0.110

TABLE VI  
RESULTS OF THE K FOLD CROSS VALIDATION ON THE MODIFIED DATA SET WITH CN + SWR AND  $10 \leq k \leq 100$ ,  $k=k+10$

classification algorithms in the context of our modified corpus.

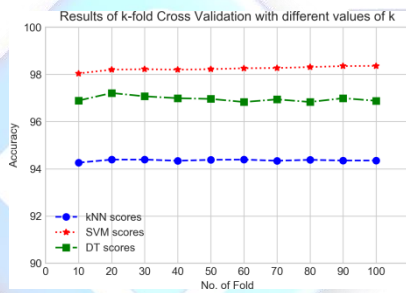


Fig. 13. Distribution of mean of mean accuracy for large values of k up to 100, using modified corpus processed by CN and SWR

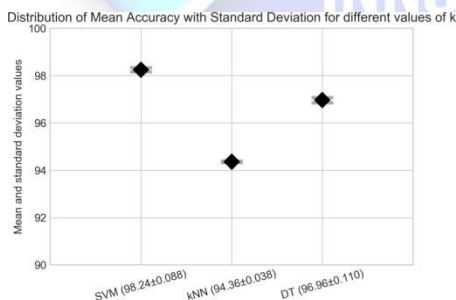


Fig. 14. Distribution of mean of mean accuracy and standard deviation for 3 classifiers with large values of k, on modified corpus processed by CN and SWR

The Table VI shows the mean of mean accuracies for the evaluation upto 100 folds for each classifier, along with the standard deviation values. The corresponding mean accuracy for every 10-fold interval is illustrated in the Figure 13, whereas Figure 14 shows the distribution of mean of mean accuracy and standard deviation for all cases.

**Observations:** From the mean of mean accuracies for the classifiers, it is seen that SVM has the highest score, followed

by DT and kNN. This is a reflection of the trend that has been seen all throughout the experiments with 10 fold cross validation on the data. Also, in this case of the modified corpus processed with CN and SWR, it is found that there is a fractional increase in the mean accuracy for both SVM and kNN in the mean of 100-fold computation. The monte carlo approach followed in all the experiments, helps to establish that SVM is a very robust learning algorithm for such text based problems. This relates with the findings of previous works by Almeida *et al.* [7] and Joachims *et al.* [21].

## VIII. CONCLUSIONS AND FUTURE WORK

In this work, spam recognition has been performed with the use of machine learning algorithms on processed and vectorized data. The text corpus has been used in its original form, and also with the inclusion of crowd-sourced Indian spam and ham SMS messages. Text processing has been done using case normalisation and stop word removal, both individually and in combination. The TF-IDF vectorized data is then evaluated with three different classifiers for both the sets of data. In order to determine the robustness of classifiers, the authors have used a k-fold cross validation, extended upto  $k=100$ . From the experiments, it has been observed that SVM performs very consistently with an accuracy rate above 94% all throughout, even in the case of the text corpus with the Indian spam messages. The kNN classifier has given the least accuracy in most cases, which may be due to the standard value of k used in our experiments, and the determination of an optimal value of k is a subject of further experimentation. The implementation of such accurate spam detection on the smart-phone devices at user end is a challenging problem, along with the context of identifying SMS messages in regional languages, typed in English font. Both of these remain interesting future works that the authors wish to pursue.

## IX. ACKNOWLEDGEMENT

The authors take this opportunity to thank their University colleagues who provided the spam and ham text message data that has been used in this work.

## REFERENCES

- [1] B. N. W. Edition, "Hppy bthdy txt!" [http://news.bbc.co.uk/2/hi/uk\\_news/2538083.stm](http://news.bbc.co.uk/2/hi/uk_news/2538083.stm), 2002, [Online; accessed 15-August-2019].
- [2] S. of Digital Formats: Planning for Library of Congress Collections, "Short Message Service (SMS) Message Format," <http://www.loc.gov/preservation/digital/formats/fd/fdd000431.shtml>, 2015, [Online; accessed 12-August-2019].
- [3] R. J. TRAPPEY III and A. G. Woodside, "Consumer responses to interactive advertising campaigns coupling short-message-service direct marketing and tv commercials," *Journal of Advertising Research*, vol. 45, no. 4, pp. 382–401, 2005.
- [4] PortioResearch, "Mobile Messaging Futures 2014-2018," <https://web.archive.org/web/20151208180248/http://www.portioresearch.com/en/messaging-reports/mobile-messaging-research/mobile-messaging-futures-2014-2018.aspx>, 2014, [Online; accessed 15-August-2019].
- [5] Q. I. C. Inbox, "The SMS inbox on Indian smartphones is now just a spam bin," <https://qz.com/india/1573148/telecom-realty-firms-banks-send-most-sms-spam-in-india/>, 2019, [Online; accessed 27-August-2019].
- [6] S. Dixit, S. Gupta, and C. V. Ravishankar, "Lohit: An online detection & control system for cellular sms spam," *IASTED Communication, Network, and Information Security*, 2005.
- [7] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*. ACM, 2011, pp. 259–262.
- [8] K. Yadav, P. Kumaraguru, A. Goyal, A. Gupta, and V. Naik, "Smsassassin: crowdsourcing driven mobile-based system for sms spam filtering," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. ACM, 2011, pp. 1–6.
- [9] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 1. IEEE, 2011, pp. 101–105.
- [10] M. Taufiq Nuruzzaman, C. Lee, M. F. A. b. Abdullah, and D. Choi, "Simple sms spam filtering on independent mobile phone," *Security and Communication Networks*, vol. 5, no. 10, pp. 1209–1220, 2012.
- [11] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "A novel framework for sms spam filtering," in *2012 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2012, pp. 1–4.
- [12] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, "Sms spam detection using noncontent features," *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 44–51, 2012.
- [13] T. Almeida, J. M. G. Hidalgo, and T. P. Silva, "Towards sms spam filtering: Results under a new dataset," *International Journal of Information Security Science*, vol. 2, no. 1, pp. 1–18, 2013.
- [14] H. Shirani-Mehr, "Sms spam detection using machine learning approach," *unpublished* <http://cs229.stanford.edu/proj2013/ShiraniMeh r-SMSSpamDetectionUsingMachineLearningApproach.pdf>, 2013.
- [15] A. Karami and L. Zhou, "Improving static sms spam detection by using new content-based features," 2014.
- [16] A. Narayan and P. Saxena, "The curse of 140 characters: evaluating the efficacy of sms spam detection on android," in *Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices*. ACM, 2013, pp. 33–42.
- [17] S. Agarwal, S. Kaur, and S. Garhwal, "Sms spam detection for indian messages," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2015, pp. 634–638.
- [18] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter sms spam," *Future Generation Computer Systems*, 2019.
- [19] U. M. L. Repository, "SMS Spam Collection Data Set," <http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/>, 2012, [Online; accessed 15-August-2019].
- [20] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, 2004.
- [21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.