A Bangla Spam Email Detection and Datasets Creation Approach based on Machine Learning Algorithms

Ruhul Amin, Md. Moshiur Rahman, and Nahid Hossain
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
ruhulamin6678@gmail.com, moshiurrahmanb11@gmail.com, and nahid@cse.uiu.ac.bd

Abstract— **Email is one of the most imperative communication mechanisms of the 21st century. People around the world send billions of emails every single day which makes people prone to threats. Spam emails can be used for stealing things from our electric devices, blackmailing and phishing. Moreover, we receive several unwanted emails such as advertisements and offers of e-commerce websites. Due to these emails, we sometimes miss important emails. There are several approaches available for detecting spam emails in the English language and other important languages. However, there is no such spam email detection tool available in Bangla language where Bangla is one of the most spoken languages in the world and widely used on the internet nowadays. For this reason, spammers nowadays send emails in Bangla language to Bengali people so that they can avoid being filtered out which makes us vulnerable to serious threats. Therefore, we have designed and developed a spam email detection mechanism in the Bangla language. We have also constructed datasets of Bangla spam emails to train and test our system. This paper explores the use of six supervised machine learning approaches. According to the classification results, Random Forest presented the best performance with 93.60% accuracy.**

Keywords—Spam, Bangla, Email, Machine Learning

I. INTRODUCTION

Email is a fast and inexpensive way of information sharing and communication in today's world. Undesirable bulk email, or more accurately, spam email has been bringing forth enormous devastation to Internet Service Providers (ISP), internet users, and the complete internet backbone [1]. People around the world send nearly 292.3 billion emails every day in which 55% are spam emails [2]. Spam emails have been used for phishing, stealing important documents, blackmailing and several other serious crimes. Therefore, spam email detection is very important for our daily internet activity. Although there is a significant amount of research works on spam email detection have been done in English, no spam email detection mechanism ever proposed in Bangla. Thus, spammers are taking this opportunity making Bangla speaking people vulnerable. According to our study, nowadays people receiving numerous amounts of spam and unwanted emails in Bangla language. At present, available email service providers such as Gmail, Yahoo failed miserably detecting spam emails written in Bangla. Bangla (endonym Bengali) is the 7th most widely spoken language around the world with approximately 228 million total speakers worldwide [3]. The number of internet users in Bangladesh and Bengali speaking people has increased dramatically in the last decade. This motivates us to build a Bangla spam detection mechanism.

As we have mentioned earlier, a considerable amount of work had been done on spam email detection in English and other western languages. In 2014, Sarju et al. experimented with 8000 English emails to build a spam email detection mechanism and their methodology preserve an accuracy of up to 99.4% [4]. Bayati and Jabbar proposed a spam detection approach in 2015 using the Naive Bayesian (NB) classifier and they have used the English dataset CSDMC2010 spam corpus [5]. Zhiwei et al. proposed a hybrid model that provides an accuracy rate of 93.00% in English Spam emails in 2017 [6]. In 2018, Tekerek and Bay projected a Spam email detection approach based on some Machine Learning(ML) algorithms where Random Tree(RT) showed the best performance [7]. In the same year, Vijayasekaran and Rosi proposed a spam email detection approach in big data platform using NB classifier [8]. On the other hand, no work has been done on Bangla spam email detection. However, some text categorization approaches were done in Bangla which is relevant to our work. In 2014, A. N. Chy et al. proposed a Bangla news classification approach using an NB classifier based on the news code of International Press Telecommunication Council [9]. In the same year, Mandal and Sen proposed a Bangla web document categorization based on supervised learning methods in which Support Vector Machine (SVM) gives the highest accuracy of 89.14% [10]. An SVM mixed with the TF-IDF algorithm was projected by Islam et al. in 2017 to classify Bangla text documents [11]. Dhar et al. showed a comparison of performance by different classifiers for text classification in the year 2018 [12].

In this paper, we have proposed a robust Bangla spam email detection approach based on machine learning approaches. We have explored supervised ML methods, namely Multinomial Naive Bayes, Decision Tree (DT), K Nearest Neighbor (KNN), Random Forest (RF), AdaBoost, and SVM. In order to train our model, we have built a Bangla spam and unwanted promotional email train dataset with 4766 emails. We have also made an independent test dataset which consists of 850 Bangla emails. We have collected all these emails from various sources including personal emails. Our model shows significant accuracy in almost all the supervised ML algorithms. However, Random Forest shows the highest accuracy. Since there are no Bangla spam email detection tool available and Bangla spam email datasets available online, we have decided to publish our project on GitHub including train and test datasets. The originality of the work is our developed Bangla spam email training and test datasets.

The paper is organized as follows: In section II, we describe the proposed system and step by step explanations of our work and algorithm. The paper demonstrates experimental results and performance analysis in section III, while section IV concludes the paper with limitations of our system and future work.

II. Proposed Method

As we have mentioned earlier, there is no state of the art work available for Bangla spam email detection that we can compare our system or can take help which leads us to design and develop the whole system by the methodology described in this section. As we mentioned in Section I, our experiment is to classify and filter unwanted Bangla emails. Unwanted emails contain both commercial emails and pure spam emails. For example,

1. “অভিনন্দন, আপনি এই বছর আমাদের কোম্পানী তরফ থেকে সবচেয়ে বড় অফার পাওয়ার যোগ্যতা অর্জন করেছেন। আপনি পাচ্ছেন আমাদের কোম্পানীর তরফ থেকে ১ কোটি টাকার বিনাসুদে ঋণ। আপনি আজই আবেদন প্রক্রিয়া সম্পন্ন করার জন্য এই লিংক অনুসরণ করুন।”

2. “ঢাকা গাউন্ডসেল এ স্বাগতম, আপনি জেনে আনন্দিত হবেন আমরা বাড্ডায় নতুন শো-রুম খুলেছি। এখানে আপনারা পাচ্ছেন ৭৫% পর্যন্ত ছাড়। আজই ঘুরে আসুন আমাদের শো-রুম থেকে। ধন্যবাদ”

From the above examples, example 1 is a pure spam email where example 2 is just an unsought commercial email. In our approach, creating a train and test dataset was the highest priority. Since there is no Bangla spam email train or test datasets available, we have decided to build these datasets at the very beginning of our work with the highest priority. For a robust spam email detection mechanism, the size of the training dataset should be sufficiently large. Therefore, our target was to compile at least 4500 Bangla spam emails for the training dataset and around 8 00 Bangla spam emails for the test dataset. We have collected these emails by a user survey online to post their Bangla spam and unwanted promotional/offer emails including any other Bangla emails that seem unwanted to the user. The contributors are mostly from 18 to 30 years old and mostly from the computer science background. Around 50 per cent of the contributors are university students. After the collection, we have filtered out some emails which should not be included in the spam email dataset. On the other hand, we have also collected emails from around 500 email users directly from their account with their supervision. Around 45% of emails were collected via an online survey and 55% directly from user emails. Then, we needed to convert some emails written in ANSI fonts to Unicode font in our case most popular Bangla Unicode font Vrinda. In some cases, this conversion took enormous time due to lack of efficient converter. The train and test datasets both have only two classes either ‘1’(Spam) or ‘0’(Not Spam). In order to achieve higher accuracy, we have made our datasets balanced keeping a 1:1 ratio of both classes. Binary classification problems require a dataset comprising of a subgroup of positive samples, in our case ‘1’(spam) and a subgroup of negative samples in our case ‘0’(not spam). Properly:

$$E = E^{+} \cup E^{-} \tag{1}$$

Here, $E+$ and $E-$ represent ‘1’(spam) samples and ‘0’(not spam) samples respectively. Table 1 shows a summary of the size of our train and test datasets.

Table 1. Size of the train and test datasets

Dataset	‘1’ (Spam) Samples	‘0’ (Not Spam) Samples	Total
Training	2382	2384	4766
Test	425	425	850

These datasets will be freely available in our GitHub repository within a few weeks after further enhancement of the datasets; so that anybody can have the opportunity to build a better classifier based on these datasets.

Algorithm 1 demonstrates the procedure of identifying whether an inputted Bangla email is spam or not. After taking a Bangla email as input we first check whether the text given is in Unicode form, if not, we convert it to the Unicode form to be accessible by our system. Then we remove all unnecessary information such as headings “জনাব (Sir)”, “মহাশয় (Sir/Mister)”, “প্রিয় শিহাব (Dear Shihab)”, “অভিনন্দন (Congratulation)” and footers such as “বিনীত (Sincerely)”, “বিনীত নিবেদন (with due respect / Sincerely) ” , “আপনার একান্ত অনুগত (your most obedient)” etc. Finding TF-IDF [13] of these headers and footers is unnecessary since these pieces of data do not give any important information to classify the email. We also remove all additional punctuations and whitespaces from the email. Then we have generated the vector representation of the email using the TF-IDF algorithm. TF-IDF is a numerical statistic that often used for information retrieval and data mining. It has been used to demonstrate how significant a word is to a corpus or in a single document. In digital libraries around 83 per cent [14] of text-based recommendation systems use the TF-IDF algorithm. We have constructed TF-IDF vector considering unigram and bigram separately and tested them against different algorithms. It has been perceived that combined unigram and bigram representation of TF-IDF gives better results. We then pass through the vector representation of the inputted email and emails from our dataset to the classification algorithm to classify the email whether the email is spam or not.

Algorithm 1 Bangla Spam Email Classification (Abstract)

```
$IN\_EMAIL$ ← input a raw Bangla email
if $IN\_EMAIL$ is not in Unicode form then
    convert $IN\_EMAIL$ in Unicode (Bangla Vrinda font)
end if
$CLEAN\_EMAIL$ ← preprocess $IN\_EMAIL$, remove any additional information, punctuations and whitespaces.
$ST\_EMAIL$ ← apply rule-based stemmer on $CLEAN\_EMAIL$ to get common base form of words.
$TF$ ← calculate TF value from $ST\_EMAIL$
$IDF$ ← calculate IDF value from $ST\_EMAIL$
$TF - IDF$ ← calculate TF-IDF score
```

```
TF – IDF_DATA← TF-IDF representation of datasets
pass TF − IDF_DATA values to classification algorithm
get classification result of the Bangla Email IN_EMAIL
```

In this paper, several classification algorithms have been used in the experiments namely Multinomial Naive Bayes (MNB), K Nearest Neighbor (KNN), Decision Tree, Support Vector Machine (SVM), Random Forest, and AdaBoost. The MNB [15] classifier is a variant of regular Naïve Bayes classifier, and it is often used in different text classification. The KNN [16] is a lazy instance-based, simple, easy-to-implement supervised machine learning algorithm. KNN can also be used in the regression. Decision tree [17] classifier best suited in binary classifications. It generates decision trees in both classification and regression where attributes are used hierarchically to find the labels of the samples as leaves. SVM [18] is a maximum margin classifier that outputs an optimal hyperplane which classifies new samples. It uses the following decision rule,

$$h(\vec{x}) = sign(\textstyle\sum_j a_j\, y_j(\vec{x}.\vec{x}_j) - b) \qquad (2)$$

Here, $\vec{x}_j$ is a support vector that defines the maximum margin. Random Forest [19] consists of multiple random decision trees that provide a classification for input data to classify a new instance. AdaBoost [20] combines several weak classifiers into a single strong classifier to classify data. The formula of the decision rule in AdaBoost classifier is as follows,

$$h(\vec{x}) = sign(a_1 h_1(\vec{x}) + a_2\, h_2(\vec{x}) + \cdot\cdot\cdot) \qquad (3)$$

Here, α_i is the weight allied with it and h_i is the weak classifier at iteration i.

We have used Random Forest as our classifier since the classifier gives the best test accuracy in our experiment explained in Section III.

III. Experimental Results and Performance Analysis

In this section, we have demonstrated the experimental results and performance analysis. The experiment was done in a high-performance computer with 32 GB RAM, 6GB Graphics Card, and Intel Xeon Processor. Python 3.6 was the core language in the experiment with the Scikit-learn library [21]. In order to validate the classifiers, we have used two methods in our experiment. First, our own independent test dataset and second, the K-Fold Cross Validation (CV) [22]. The K-Fold CV divides the entire dataset into K folds or sections. In these folds, each fold is used as a test set at some point of the entire process. In our experiment, we have used 10 as the value of K which means the entire dataset divided into 10 chunks. Each time one chunk works as a test set and all the other chunks work as train set. On each iteration, test and train sets changes automatically. The summary of the average 10-Fold score of our dataset is given in Table 2.

Table 2. Summary of average K-Fold score

Classifier	Average K-Fold Score
Multinomial NB	0.8726
kNN	0.8651
Decision Tree	0.8863
SVM	0.8817
Random Forest	0.9010
AdaBoost	0.8858

However, as we have mentioned earlier, we have our own independent test set, therefore, we have tested the trained model using the dataset also. In this experiment, we have used three performance evaluation metrics and they are Accuracy, Sensitivity or Recall, and Specificity. These metrics can be defined as follows:

$$Accuracy = (TN + TP)/(TN+TP+FN+FP) \qquad (4)$$

$$Sensitivity = TP/(TP + FN) \qquad (5)$$

$$Specificity = TN/(TN + FP) \qquad (6)$$

where:

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

The result of the experiment has been shown graphically in Figure 2 and the summary is given in Table 3.

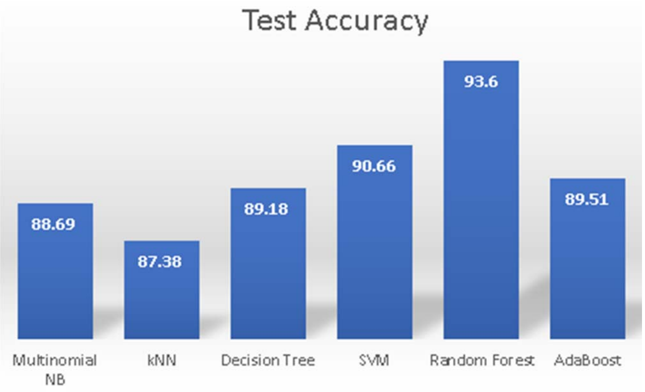

Fig. 1. Graphical representation of accuracy on independent test dataset

Table 3. Summary of experimental results

Classifier	Accuracy (%)	Sensitivity	Specificity
Multinomial NB	88.69	0.8951	0.8788
kNN	87.38	0.8689	0.8786
Decision Tree	89.18	0.8984	0.8852
SVM	90.66	0.9148	0.8984
Random Forest	93.60	0.9409	0.9311
AdaBoost	89.51	0.8885	0.9016

According to Figure 2, the Random Forest classifier shows the best performance in our experiment with an accuracy rate of 93.60 per cent. With the accuracy rate of 90.66 per cent SVM is the second-best classifier while Decision Tree, AdaBoost, and KNN have fairly good accuracy. On the other hand, Multinomial NB has the lowest accuracy in this case.

We have observed the Sensitivity and Specificity of the Random Forest are also the highest compared to other algorithms. This suggests us to use the Random Forest as our fundamental classifier of the system.

IV. Conclusion and Future Work

This paper demonstrates a comprehensive Bangla spam email detection system based on machine learning algorithms. We have experimented with several algorithms and finally decided to use the Random Forest classifier since it gives us the highest accuracy. We have built first ever Bangla spam email datasets that contain a decent number of samples and these datasets will be freely available in our GitHub repository for other researchers.

Although we have reached our goal, the system has some limitations. According to our study, the main limitation is the size of the datasets. Although this is the first dataset of its kind, we are trying to increase the size of our datasets. Moreover, in our current experiment, we have used only generic machine learning algorithms. Our next strategy is to use Deep Learning (DL) approaches such as Deep Neural Networks to make the system more robust. In the future, we are planning to do a user experience survey according to Human-Computer Interaction (HCI) discipline based on a robust online and offline questionnaire to improve the quality of the work. Since the originality of the work is mainly our developed Bangla spam email training and test datasets, we will upload the datasets along with codes into our GitHub repository for other researchers.

References

[1] Hidalgo, J. M. G. (2002, March). Evaluating cost-sensitive unsolicited bulk email categorization.In Proceedings of the 2002 ACM symposium on Applied computing (pp. 615-620). ACM.

[2] "The Surprising Reality of How Many Emails Are Sent Per Day - Tech Jury", Tech Jury, 2019. [Online]. Available: https://techjury.net/stats-about/how-many-emails-are-sent-per-day/.

[3] Ethnologue. (2019). Summary by language size. [online] Available at: https://www.ethnologue.com/statistics/size .

[4] Sarju,Riju Thomas and Emilin shyni "Spam email detection using structural featues", International Journal of computer Applications,vol 89, issue 3, March 2014.

[5] Bayati, Maha Adham and Jabbar, Saadya Fahad "Developing a Spam Email Detector", International Journal of Engineering and Innovative Technology, Volume 5, Issue 2, August 2015.

[6] ZhiWei, Mi, Manmeet Mahinderjit Singh, and Zarul Fitri Zaaba. "Email Spam Detection: A Method Of Metaclassifiers Stacking." In The 6th International Conference on Computing and Informatics, pp. 750-757. Kuala Lumpur Malaysia, 2017.

[7] Tekerek, Adem & Bay, Omer. "Spam E-Mail Detection Based On Machine Learning", 7th International Conference on Advanced Technologies, 2018

[8] G.Vijayasekaran, S. Rosi "Spam And Email Detection In Big Data Platform Using Naives Bayesian Classifier" International Journal of Computer Science and Mobile Computing, Vol. 7, Issue. 4, April 2018, pg.53 – 58

[9] A. N. Chy, M. H. Seddiqui and S. Das, "Bangla news classification using naive Bayes classifier," 16th Int'l Conf. Computer and Information Technology, Khulna, 2014, pp. 366-371.doi: 10.1109/ICCITechn.2014.6997369

[10] Mandal,Ashis Kumar and Sen, Rikta "Supervised Learning Methods For Bangla Web Document Categorization", International Journal of Artificial Intelligence & Applications, Vol. 5, No. 5, September 2014

[11] M. S. Islam, F. E. M. Jubayer and S. I. Ahmed, "A support vector machine mixed with TF-IDF algorithm to categorize Bengali document," 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, 2017, pp. 191-196.doi: 10.1109/ECACE.2017.7912904

[12] Dhar, Ankita & Mukherjee, Himadri & Dash, Niladri & Roy, Kaushik. "Performance of Classifiers in Bangla Text Categorization", International Conference on Innovations in Science, Engineering and Technology 2018, 10.1109/ICISET.2018.8745621.

[13] Qaiser, Shahzad & Ali, Ramsha. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications. 181. 10.5120/ijca2018917395.

[14] Breitinger, Corinna; Gipp, Bela; Langer, Stefan (2015-07-26). "Research-paper recommender systems: a literature survey". International Journal on Digital Libraries. 17 (4): 305–338. doi:10.1007/s00799-015-0156-0. ISSN 1432-5012

[15] Xu, Shuo & Li, Yan & Zheng, Wang. (2017). Bayesian Multinomial Naïve Bayes Classifier to Text Classification. 347-352. 10.1007/978-981-10-5041-1_57.

[16] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. MIT press, 2012.

[17] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3):660–674, 1991.

[18] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

[19] Tin Kam Ho. Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on, volume 1, pages 278–282. IEEE, 1995.

[20] Robert E Schapire. The boosting approach to machine learning: An overview. In Nonlinear estimation and classification, pages 149–171. Springer, 2003.

[21] Fabian Pedregosa, Ga¨el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikitlearn: Machine learning in python. Journal of Machine Learning Research, 12(Oct):2825–2830, 2011

[22] Jung, Yoonsuh & Hu, Jianhua. (2015). A K-fold Averaging Cross-validation Procedure. Journal of Nonparametric Statistics. 27. 1-13. 10.1080/10485252.2015.1010532.