Identification of Extreme Guilt and Grave Fault in Bengali Language using Machine Learning

Aloke Kumar Saha, Jugal Krishna Das

Abstract: ***Though huge amount of study has been done on the Bengali Language for information retrieval, but none of them deals with extreme guilt (বাহুল্য দোষ) and grave fault (গুরুচন্ডালী দোষ) in the Bengali Language. In this study, we have described extreme guilt (বাহুল্য দোষ) and grave fault (গুরুচন্ডালী দোষ). We have used three machine learning methods, such as Logistic Regression (LR), Support Vector Machine (SVM) and Multinomial Naive Bayes (MNB) as the baseline classifiers among the baseline classifier, MNB shows the accuracy of 89%. Ensemble learning has been used to improve the baseline classifiers. We have implemented an Ada Boost algorithm and Maximum voting classification decision method depending on the results of baseline classifiers. Maximum voting and Ada-Boost algorithms have shown an accuracy of 91% and 92% respectively. We have modified the Ada-boost algorithm using Principal Component Analysis (PCA) and named it JR-Ada-Boost. It outperforms all algorithms and gives an accuracy of 94%.***

Keywords: ***extreme guilt · grave fault · information retrieval · text analysis · data analysis · artificial intelligence.***

I. INTRODUCTION

To construct a substantial (সার্থক) Bengali sentence, it is important to have a valid combination between emotion (ভাবগত) and meaning (র্অথগত) of the words in a sentence [1]. Some subject matter is related with this compatibility (যোগ্যতা)
issue. Some of them are:

– Extreme Guilt (বাহুল্য দোষ)

– Grave Fault (গুরুচন্ডালী দোষ)

Revised Manuscript Received on February 18, 2020.

Aloke Kumar Saha Department of CSE, University of Asia Pacific, Dhaka, Bangladesh, Email: aloke@uap-bd.edu

Jugal Krishna Das, Department of CSE, Jahangirnagar University, Savar, Dhaka, Bangladesh, Email: drdas64@yahoo.com

Table 1 Examples of Extreme Guilt (বাহুল্য দোষ)

Sentence	Comment
(ক্লাসরে সব ছাত্রবৃন্দ আজ উপস্থিতি) All students are present in today's class	False
(ক্লাসরে সব ছাত্রই আজ উপস্থিতি) All student are present in today's class	Correct
(সে আমার কাছে আরও প্রিয়তর) He's even more darling to me	False
(সে আমার কাছে আরও প্রিয়) He's even more dear to me	Correct

Table 2 Examples of Plain (সাধু) & Prevailing (চলতি) Language

Sentences	Comments
(তমার কাছ পাইয়া আমি ধন্য হইলাম) I am blessed to have you	Plain (সাধু)
(তোমাকে কাছে পেয়ে আমি ধন্য হলাম) I'm blessed to have you	Prevailing (চলতি)

1.1 Extreme Guilt (বাহুল্য দোষ) Extreme Guilt (বাহুল্য দোষ) occurs, when a sentence has unnecessary words or adjectives or modifiers. Examples are given in Table 1. In Table 1, there are two examples of extreme guilt (বাহুল্য দোষ). The first sentence has a word (ছাত্রই) (students) in it. According to Bengali grammer [1], if a sentence has a word (সব)(All), than that sentence can't have (ছাত্রই) (students). So, the correct sentence must have the word (ছাত্রই) (student) in it. The second sentence has a word (প্রিয়তর) (darling) in it.

According to Bengali grammer [1], if a sentence has a word (আরও)(more), than that sentence can't have (প্রিয়তর) (darling). So, the correct sentence must have the word (প্রিয়) (dear) in it

1.2 Grave Fault (গুরুচন্ডালী দোষ) There are two variants of Bengali language[1]. They are – Plain Language (সাধু ভাষা) – Prevailing Language (চলতি ভাষা) The language, which completely adheres to the rules of grammar is called plain language (সাধু ভাষা) and different regionally used or verbal languages are called prevailing languages (চলতি ভাষা). Prevailing Languages (চলতি ভাষা) don't follow any grammatical rules[1]. Examples are given in Table 4. Significant differences exist between the two methods. So, the combination of plain (সাধু) and prevailing (চলতি) is invalid in the Bengali Language. This blatant application of linguistics is a grave fault (গুরুচন্ডালী দোষ). Examples are given in Table 3. In Table 3, there are two sentences. First one is (ঘোড়ার গাড়ী) (horse cart) is a prevailing language (গুরুচন্ডালী দোষ) but in case of (ঘোড়ার শকট) (horse cart) is not a prevailing

Table 3 Examples of sentences consists of grave fault (গুরুচন্ডালী দোষ)

Sentence	Comment
(ঘোড়ার শকট) horse cart	False
(ঘোড়ার গাড়ী) horse cart	Correct
(শবপোড়া) Bury	False
(শবদাহ) Bury	Correct

language (চলতি ভাষা) nor a plain language (সাধু ভাষা). Here, (ঘোড়ার) is a prevailing language (চলতি ভাষা) and (শকট) plain language (সাধু ভাষা). Second sentences follows the same rule. For feature extraction, we have used CountVectorizer [2], Tf-Idf [3] and Word2Vec [4]. For Machine classification Multinomial Naive Bayes [5], Support vector Machine [6] and Logistic Regression [7] were implemented and results were compared. The best result has been selected as the benchmark. These three machine learning classifier algorithms are used in maximum voting and also for ensemble learning called Boosting [8]. Rest of the paper is articulated as follows. Section 2 presents the related works. Methodologies are described in section 3. Result analyses are discussed in section 4. We conclude our paper in section 5.

II. RELATED LITERATURE

Lee et al [9] proposed a model, where the frequency of keyword was count from a text. All frequencies and features were given as the input of the (KSOM) Korhonen selforganizing neural network. It is used to identify the pornographic text. Sassanoy [10] had proposed methods or process to create virtual examples on the assumption that the label of a document remains the same even if a small number of words are added or deleted. Although the proposed methods do not apply to NLP tasks other than text classification, it is notifying that the use of virtual examples, which has been very little studied in NLP, is extremely evaluated. Du et al[11] focused on pornographic and general text. An algorithm mainly used for decision making in text classification is used to classify offensive texts. ÖlveckÝ and Tomáš [12] selected a text-based African language. They also selected Naive Bayes and SVM classifier. They evaluate different factors to influence LID accuracy at the time of extracting generic words. Lee et al [13] proposed a dual linguistic web page classification software. That software can identify whether a web page contains pornographic materials for a Chinese or English language or not. They had developed the classification software to execute analysis in the offline web page and online filtering in near-instantaneous. Xiang et al [14] proposed a statistical model-based method on a large corpus of Twitter data to detect profane tweets using linguistic regularities in the offensive text. Sood et al [15] proposed a method using lists and edit distance metrics and also used crowdsourcing, which an annotate abusive language. The outcome shows an improvement over previous methods. Warner and Hirschberg [16] investigate and analyzed hate speech to date. He also focused on annotation tasks and practical explanations. They mainly focused on anti-Semitic hate than on profane language. Sood et al [17] analyzed the comment from a different site of social news. They investigate and found that the performance of the ongoing process is very poor. They also identify the situations, where the current process are failed to progress. Authors proposed a better profane detection method after identifying social differences based on the tolerance of profanity. Lee and Dernoncourt [18] worked on building a sequential short-text classifier. Their classifier employed the use of a recurrent neural network. They prove that their model achieves state-of-the-art results on three different data-sets for dialog act prediction. Nobata et al [19] stated that Yahoo has governed online violations. They train a classifier with the combination of lexical, syntactic and parser features using supervised machine learning. Nobata et al [20] tried to identify the hate speech from the comments of internet users. Authors proposed a machine learning approach. Current deep-learning approaches are outperformed by their machine learning method. Authors have used their software to determine abusive text. Chu et al [21] applied Deep Learning method and Neural Language Processing to detect & classify abusive comment using a CNN with character embedding and LSTMs with word embedding. They tested three models: first is, recurrent neural network (RNN) with a long-short-term memory cell (LSTM) and word embedding, second is, convolutional neural network (CNN) with word embedding, and third is CNN with character embedding. Haque et al [22] worked on our topic with a small amount of dataset and used two machine learning methods, such as Logistic Regression (LR) and Support Vector Machine (SVM).

He got 80% and 87% of accuracy for extreme guilt (বাহুল্য দোষ) respectively and 60% and 85% accuracy for grave fault (গুরুচন্ডালী দোষ) respectively. From the above discussion, we can conclude that there were is only a few works has done on our focused topic with small data set. None of the studies developed ensemble learning and also didn’t focus on varieties of Bengali Offensive texts. In our research, we have implemented ada-boost with the combination of baseline classifiers and modified it using Principal Component Analysis (PCA).

III. METHODOLOGY

In this section, we are going to discuss our data and algorithms we have used for this study. For this study, we have implemented machine algorithms like SVM with linear kernel, Logistic Regression, MNB for the baseline classification. For feature extraction, we have used the Tf-idf vectorizer, count vectorizer, and word2vec. Again, we have developed the two most popular ensemble method named as multi voting and Boosting (Ada-Boost) using those baseline classifiers. Finally, we have discussed our proposed method. We have modified the Boosting (Ada-Boost) method using PCA (Principal component analysis). To reduce the size of our dataset and also to focus on the vital portion of the data set, we have used PCA.

3.1 Data collection

It is better to collect data from different social network platform than manually developing a list of sentence to represent Bengali offensive text. All social media platform doesn’t have Bengali language users so, we have gathered comments and posts from different Bengali websites :1 and social media platforms : 2 3 . From Bengali websites, blogs and different social media platforms, we have collected 1500 posts and comments. All posts and comments that we have collected are under a limited size. We discarded all the links, date, time and user information for better accuracy at the time of machine classification and also for user privacy. We removed punctuation’s, symbols, special characters, and other languages from the data-sets. We labeled all the data sets. From, the data-set, we kept 30 percent of the total dataset for testing and 70 percent of the total data-set for training. We have collected the data year wise and we also kept track of the platforms from which that data belong. User information, date, time were discarded for user privacy issue. We have also discarded the special characters, symbols and signs. All other language expect Bengali were discarded.

3.2 Feature extraction

For feature extraction we have used CountVectorizer [2], Tf-Idf [3] and Word2Vec [4].

3.2.1 Countvectorizer

CountVectorizer mainly used to convert any text into the vector. Some’s it’s also for the extraction of features. A matrix of the token has been created by converting a set of text documents. A sparse presentation of the counts is implemented using scipy.sparse.csr_matrix. An a-priori dictionary is not provided and the analyzer is not used. Only after that, two-part it acts like feature selection. After data analysis, the feature number should be equal to the size of the vocabulary.

3.2.2 Tf-Idf vectorizer

At the time of processing a large text data-set, there are some words which occur frequently but possess insufficient essential information. Those data generally shadow the frequency of more significant data at the time of text processing with the direct count. To solve this issue, it is very common to use tf-idf. Tf-idf means term-frequency times inverse document - frequency.

$$\mathrm{tf} - \mathrm{idf}(t, d) = \mathrm{tf}(t, d) \times \mathrm{idf}(t) \qquad (1)$$

Where, tf(t,d) is the term frequency and idf(t) is the inverse document frequency.

3.2.3 Word2vec

It has been used to determine the vector of all terms in a manner, where the alike text has related vectors. It can be used in semantic analysis of texts. It is a two-layer neural network. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. It is not a deep neural network

3.3 Data set pre-processing

Three machine classification algorithms. Data preparation is done with word2vec and count vectorizer for MNB and logistic regression respectively. Tf-idf Vectorizer has been used for SVM with Linear Kernel. To achieve our intended goal of offensive text detection, we have explored two very important issues. Firstly, the exploration of character embedding instead of word embedding exploiting countvectorizer. Rigorous investigation on the pre-processed data reveals that in many places, words did not conform to dictionary words and mixed special characters with words. This happens especially in ‘abuse’ comments. Character embedding can do better in capturing this issue. Secondly, a very popular approach called ‘fast Text’ has also been explored and it has been proved to be extremely effective in text classification. In this paper, we have implemented the first method for offensive text detection. We plant to delve the later issue in future works.

3.4 Machine classification

3.4.1 Baseline experiment

Baseline experiment has been conducted using python machine learning libraries. We have used binary classification process using 3 most popular machine learning classifiers in text classifications. These were Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naive Bayes (MNB). Above mentioned classifiers have been used for some reasons. SVM shows better performance with small amount of data [23,24]. It also shows remarkable outcomes at the time of classifying mental health issue [25]. All featured vectors have been plotted within a higher dimensional space. Hyperplanes search the minimal steps to separate the space in a way that the point belongs to the different class are divided. Sometimes, more than one hyperplanes are used with the goal of finding the target hyperplane, which can maximize the distance among classes. Logistic Regression is very useful at the time of classification. It estimates all connection among all certain independent and dependent variable by considering possibilities applying logistic functions. We executed the Multinomial Naive Bayes algorithm as a probabilistic method.

3.4.2 Ensemble experiment

All particular baseline classifiers show reasonable output but those outputs must need to be filtered. In section 4, we are going to discuss it further. This refers that

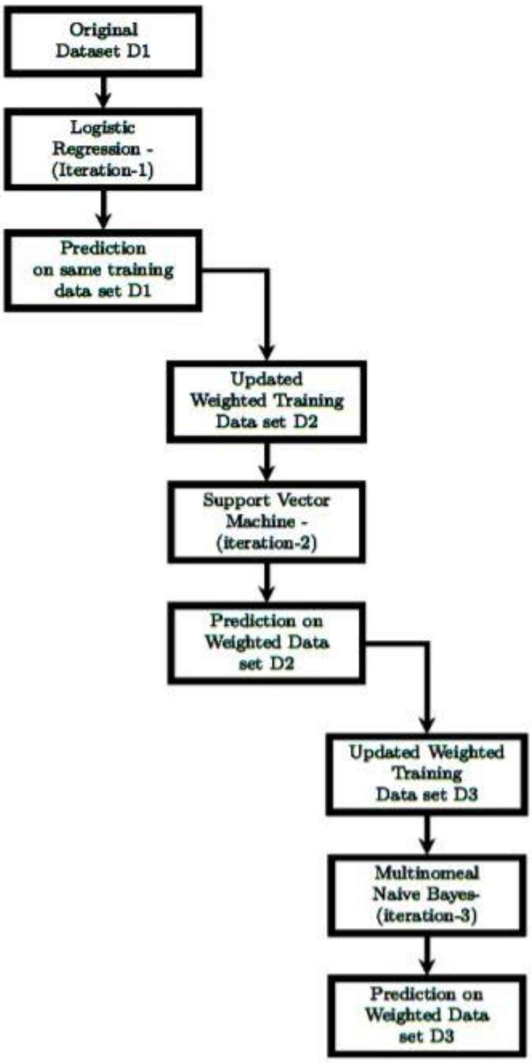

Fig. 1. Ensemble learning-boosting (Ada-boost)

the data set used wasn't worth enough to allow the baseline classifiers to properly learn a standard group of features. It may also indicate that the binary features were not adequate for presenting the latest meaning of the offensive text or normal text. Maybe the binary features were not utilized properly at the time of the learning phase. Due to the problem with date size and binary features set bindings forced us to go for ensemble classifiers. Ensemble classifiers allow combining the baseline classifiers and other different methods for sampling the features at the time of learning stage. Most popular ensemble approaches are Boosting and Bagging [26]. A new ensemble method proposed in [27]. That method is known as Rotation Forest. RF divide all feature groups into a number of tiny sets. Principal Component Analysis is analyzed for every small set. Each subset of feature has a unique principal component of its own. The total process has been used subsequently for setting a number of classifiers. RF is not a popular ensemble algorithm but sometimes it shows better performance than Bagging and Boosting. Bagging, it is the second most popular ensemble method. It holds a part of the sample of data as a point. It trains the classifiers for all sample of the data point. It finds out the probabilities and averages them. This process goes on for every single class over all the baseline classifiers in the ensemble methods. The most popular ensemble method is Boosting. One of the examples of Boosting is Ada-Boost. Its goal is to boost the outcomes of the classifiers. A new classifier is added to the ensemble method on iteratively. Each of the newly added classifiers is practiced with the data for which the performance of the classifiers was poor in the previous iteration. Boosting has both pros and cons. It makes easy to deal with the difficult instances, which are hard to classify. It can improve the performance of the classifiers but, it is mainly applicable for the smaller dataset. As the new classifiers are forced to deal with the challenging points, sometimes some data points could be sacrificed. It may reduce the accuracy of the classifiers. Fig. 1 shows the flow-chart of ensemble method called Boosting (Ada Boost). The maximum voting classification decision method is a simple Algorithm. All baseline classifiers are used for classification. As our data is composed of binary features, we used all the baseline classifiers for certain data. The class for which the classifiers will vote most that data will belong to that class. Suppose, for a particular text in Fig. ??, SVM says the text is offensive. For that same text Logistic regression says that the text is a normal text. Again, Multinomial Naive Bayes says that the text is an offensive text. Here, out of three classifiers, two of them says that the text is offensive. Clearly, the offensive class got the maximum number of vote. So, this is recognized as an offensive text.

3.5 JR-Ada-Boost

We have implemented and modified the Ada-boost algorithm. We have added Principal component Analysis (PCA) before the data-set has been transferred for the classifiers. People generally used PCA to reduce the dimensions of a larger data-set. Fig. 2 shows the flow chart of our proposed model. Principal Component Analysis a general mathematical study, which is employed to find the high variation in a data-set. It is an unsupervised dimensionality compression method. It doesn't use.

Table 4 Performance comparison

Classifiers	TP	TN	FP	FN	Parameter Accuracy	Precision	Recall	F1-Score
LR	3900	233	600	307	82%	87%	93%	0.89
SVM	4000	284	500	256	85%	88%	94%	0.90
MNB	4100	389	400	156	89%	91%	96%	0.93
M Voting	4150	437	350	103	91%	92%	97%	0.94
Ada Boost	4245	392	255	148	92%	94%	96%	0.95
JR-Ada Boost	4310	428	190	112	94%	96%	97%	0.96

the label indicated in the data-set. For our study, we used PCA to identify the column, which has the most variation and remove the unnecessary column. Our whole dataset is not trained at a time as we used separate baseline classifiers for ensemble learning. We used PCA to identify the most variant portion of our data-set. Fig. 2 shows that, before we send the data to logistic regression, we have applied PCA to identify the most variant portion of our dataset. As our data set is divided into 3 portions for 3 classifiers, so we need to find the most variant portion among the 3 portions. The most variant portion of the data set must be the first to go for the classifiers.

IV. RESULT ANALYSIS

4.1 Parameters for performance analysis

We measure the performance of our system based on Accuracy, Precision, Recall and F1-score.

4.1.1 Accuracy

It is a mathematical measurement. It indicates that how better a classifier accurately classifies or prohibits a condition. It is known to be the symmetry of actual outcomes among the cumulative amount of samples tested.

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ (2)

where: TP = True positive; FP = False positive; TN = True negative; FN = False negative.

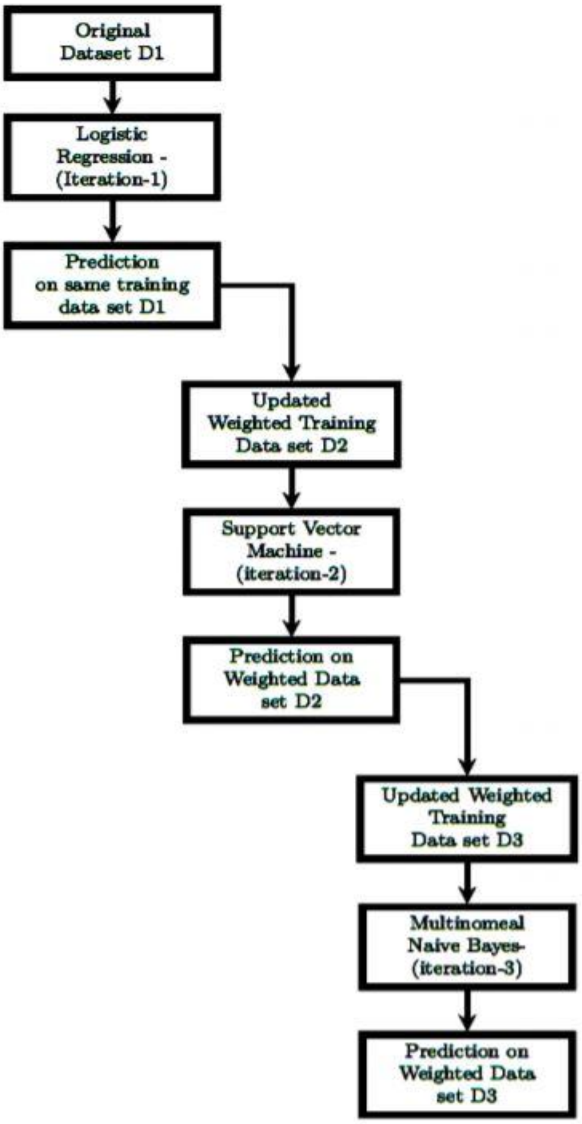

Fig. 2. Proposed (JR-Ada-boost) model

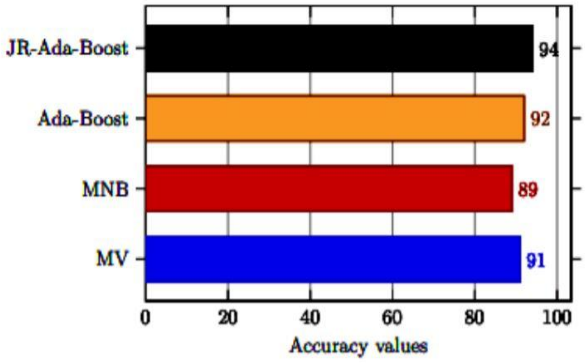

Fig. 3. Accuracy of the systems

4.1.2 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = $\frac{TP}{TP+FP}$ (3)

4.1.3 Recall

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

Recall = $\frac{TP}{TP+FN} = \frac{TP}{P}$ (4)

4.1.4 F1 score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1–Score = $2 \times \frac{Precession \times Recall}{Precession+Recall}$ (5)

4.2 Performance Analysis

We have used 70% and 30% of the total data set for the training and testing of the classifiers respectively. We have used SVM, LR, MNB machine classifiers for baseline classification and multi voting, boosting has used for ensemble learning. Table 4

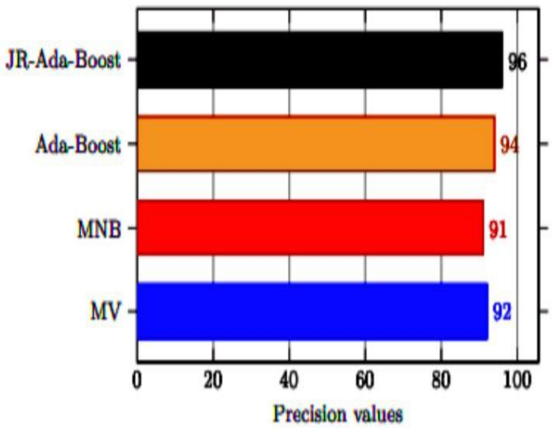

Fig. 4. Precision of the systems

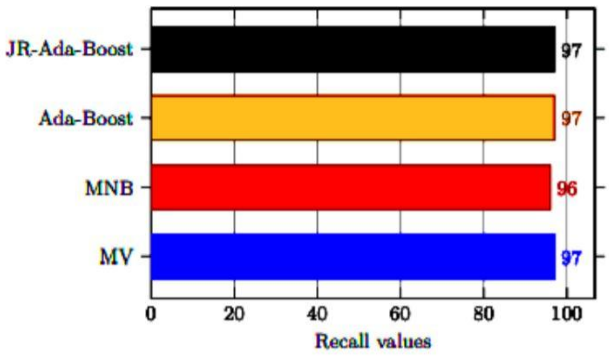

Fig. 5. Recall of the systems

shows the performance of all baseline classifiers and ensemble learning classifiers. Table 4 shows that MNB has the benchmark results from the baseline classifiers. It also has an accuracy and precision value of 89% and 91% respectively. The better precision value indicates that the classifiers have correctly classified most of the offensive text as the offensive text. Taking the vote from all the baseline classifiers, we decide the class of the data. Multi voting classification algorithm gives an accuracy of 91%. It outperforms the best baseline classification method for this data set. We have used an approach of 10 cross-validations for the evaluation of our ensemble classification system. All classifier are trained iteratively in this procedure. We have used the same amount of data set for the testing and training of the ensemble approach that has been used previously for the baseline classifiers. After 10 iterations, we have calculated the results. We took the accuracy, precision, and recall of the last model as Boosting works sequentially. The results are shown in Table 4. Ada Boost showed an accuracy of 92%. Better accuracy means that the system has correctly detected most of the offensive text and also ignored the normal text. Here, accuracy is showing an all-over performance. We

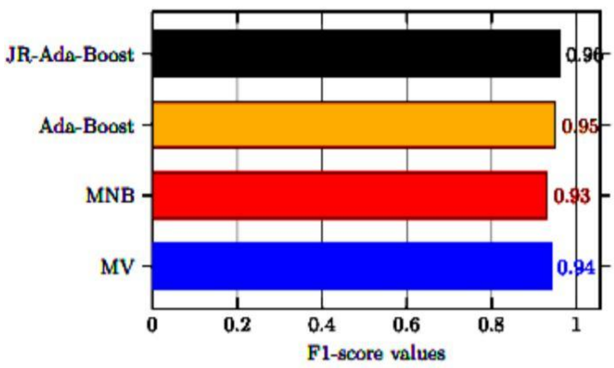

Fig. 6. F1-score of the systems

should check precision and recall values for a better understanding. The precision of our system is 94%. Of the text classified as offensive, 94% were actual offensive text. The recall value of the ensemble system is 97%. Of the actual offensive text, 97% were classified as offensive text. Our proposed JR-Ada-Boost shows an improvement of 2% than general Ada-Boost. JR-Ada-Boost shows an accuracy of 94%. Fig. 3, 4, 5 and 6 show the bar charts of all the parameters for MNB, Maximum Voting classification, Ensemble learning (Boosting) and our proposed method JR-Ada-Boost. MNB shows the best accuracy 89% among baseline classifiers. We got an improvement of 2% when maximum voting shows an accuracy of 91%. In our baseline experiment. Again, we have got an improvement of 1%, when Ada-Boost shows an accuracy of 92%. JRAda-Boosting outperforms all others Baseline, maximum voting, general Ada-Boost algorithms and shows an accuracy of 94%. Now we will compare our effort with some other works. Foqrul et al [?] acted over 1500 Bengali sentences and classified them with SVM (linear Kernel) with a feature called TfidVectorizer. He got 98.70% of accuracy. We have implemented the same method that Foqrul et al [?] used in his experiment with 16800 data sets. We got an accuracy of 85%. In Fig. 3, we showed the comparison in a graphical view

V. CONCLUSION

In this research, we developed 3 baseline machine classifiers. Our baseline classifiers shows outstanding performance. We modified those machine classifiers into a ensemble learning which shows far more better performance than any work, which were done in offensive text detection. There are several limitations present in our research. Our data set for the case studies is not diverse enough to have a better view. Bengali people of different place speaks in different versions of Bengali language. In the future, we are going to develop a deep learning model to detect Bengali offensive text. We will also analyze this topic with a larger and diverse data set to have a better view on Bengali offensive text.

REFERENCES

1. Hayat Mahmud and Muhammad Amin, Bangla Vasha and Nirmity, Puthiniloy,(2016).
2. CountVectorizer: https:// scikit - learn.org / stable/ modules/ generated/ sklearn.feature _ extraction . text . CountVectorizer . html
3. Tf-Idf: https: // scikit - learn . org / stable / modules / generated / sklearn . feature _ extraction . text . TfidfVectorizer . html
4. word2vec: https: // radimrehurek . com / gensim / sklearn _ api /w2vmodel . html
5. Multinomial Naive Bayes: https:// scikit-learn . org / stable / modules / generated / sklearn . naive _ bayes . MultinomialNB . html
6. Support Vector Machine: https: // scikit-learn . org/ stable / modules / generated / sklearn . svm . SVC . html
7. Logistic Regrassion: https: // scikit - learn.org /stable / modules / generated / sklearn . linear _ model . LogisticRegression . html
8. Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the Computational Learning Theory, Springer, pp. 23–37, (1995).
9. P. Y. Lee, S.C. Hui, A.C.M. Fong, Neural networks for web content filtering, IEEE Journal on Intelligent Systems, Vol. 17, Issue 5, pp. 48-57, 2002.
10. Manabu Sassanoy, Virtual Example for Text Classification with Support Vector Machine, Published in EMNLP 2003, DOI:10.3115/1119355.1119382
11. R. Du, R. Safavi-Naini, W. Susilo, Web filtering using text classification, IEEE International Conference on Networks, pp. 325-330, 2003
12. ÖLVECKÝ, Tomáš, N-Gram Based Statistics Aimed at Language Identification, IIT. SRC, 2005.
13. P. Y. Lee, S. C. Hui, A.C.M. Fong, An intelligent categorization engine for bilingual web content filtering, IEEE Transactions on multimedia, Vol. 7, Issue 6, pp. 1183 - 1190, 2005.
14. Guang Xiang, Bin Fan, Ling Wang, Jason Hong, Carolyn Rose, Detecting offensive tweets via tropical feature discovery over a large-scale twitter corpus, 21st ACM International Conference on Information Knowledge Management, pp. 1980 - 1984, 2012.
15. S. O. Sood, J. Antin, E. F. Churchill, Using crowdsourcing to improve profanity detection, In AAAI Spring Symposium, Wisdom of the Crowd, 2012

16. W. Warner, J. Hirschberg, Detecting hate speech on the world wide web, In Proceedings of the Second Workshop on Language in Social Media, June 2012.
17. S. Sood, J. Antin, E. Churchill, Profanity use in online communities, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1481-1490, ACM, 2012
18. Ji Young Lee, Franck Dernoncourt, Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, March 2016
19. Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, Yi Chang, Abusive Language Detection in Online User Content, authors site if the Material is used in electronic media, The International World Wide Web Conference Committee (IW3C2), WWW 2016, April 11–15, 2016, Montreal, Québec, Canada. ACM 978-1-4503-4143-1/16/04.
20. Chikashi Nobata, Joel R. Tetreault, Achint Oommen Thomas, Yashar Mehdad, Yi Chang, Abusive Language Detection in Online User Content, WWW, 2016, DOI:10.1145/2872427.2883062.
21. Tse-hung Chu, Kylie Jue, Max L. Wang, Comment Abuse Classification with Deep learning, Chu2017CommentAC, 2017
22. Rakib Ul Haque, M. F. Mridha, Aloke Kumar Saha, Md. Abdul Hamid, Kamruddin Nur, Identification of Extreme Guilt and Grave Fault in Bengali Language,ICCA 2020, January 10–12, 2020, Dhaka, Bangladesh
23. A. Pak, P.Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, in: Proceedings of the LREC, 2010
24. C. Yang, K.H. Lin, H.-H. Chen, Emotion classification using web blog corpora, in : Proceedings of the Web Intelligence, IEEE/WIC/ACM International Conference on, IEEE, 2007, pp. 275-278
25. M. De Choudhury, M. Gamon, S. Counts, E. Horvitz, Predicting depression via social media., in: Proceedings of the ICWSM, 2013.
26. L. Breiman, Bagging predictors, Mach, Learn. 24 (2) (1996) 123-140
    J.J. Rodriguez, L.I.Kuncheva, C.J. Alonso, Rotation forest: A new classifier ensemble method, IEEE Trans. Patt. Anal. Mach. Intell. 28 (10) (2006) 1619-1630.