Multilingual SMS Text Classification for Spam and Ham Detection using Natural Language Processing

Kawshik Kumar Ghosh, Ali Asgar Tamjid, Sadid Islam, Esaba Ahnaf Ibrahim,
Sifat E Jahan, Annajiat Alim Rasel
Department of Computer Science and Engineering
Brac University, Dhaka, Bangladesh
Email: kawshik.kumar.ghosh@g.bracu.ac.bd, ali.asgar.tamjid@g.bracu.ac.bd , sadid.islam@g.bracu.ac.bd,
esaba.ahnaf.ibrahim@g.bracu.ac.bd, sifat.jahan@bracu.ac.bd, annajiat@bracu.ac.bd

***Abstract*—This paper addresses the problem of spam promotional SMS messages and proposes a machine learning and deep learning-based approach for detecting them accurately. The use of SMS as a communication tool has increased, leading to a rise in unsolicited promotional messages that can be intrusive and frustrating for users. To develop and evaluate text classification models for spam detection in a multilingual context, a dataset of $2,876$ SMS messages written in Bangla, English, and Bangla-English mixed languages were used. On the dataset, many machine learning and deep learning models were used to detect spam SMS texts. Leveraging Natural Language Processing, the proposed approach offers a potential solution to the problem of spam promotional SMS messages, providing a means to filter out unwanted messages and improve the SMS messaging experience for users. After the classification, Random Forest had the highest accuracy of 93.51%. The accuracies of the Support Vector Machine as well as Multi-Layer Perceptron Classifier were 86.79% and 85.63%, respectively. Logistic Regression had the highest accuracy of 76.25%, while Multinomial Naive Bayes had the lowest accuracy of 74.62%. The dataset provides a representative sample of real-world text messages in a multilingual context, allowing for the development and evaluation of text classification models for spam detection in different languages.**

***Index Terms*—Natural Language Processing, Machine learning, Deep learning, Spam, Promotional SMS messages, Multilingual, Random Forest, Support Vector Machine, Multi-Layer Perceptron Classifier, Logistic Regression, Multinomial Naive Bayes, Real-world text messages.**

I. Introduction

The increasing use of SMS messages as a means of communication has led to a rise in the problem of spam promotional messages. These unsolicited messages can be intrusive and frustrating for users and often contain advertisements for products and services, leading to missed opportunities for important messages. In recent years, this issue has become more pronounced, and it is essential to address it. Short message service (SMS) usage provides businesses with a never-before-seen opportunity to engage with clients, and 48% of consumers prefer SMS for direct messaging, according to [9]. However, customers are also susceptible to SMS attacks like spam and phishing, especially if they are not aware of the dangers of the internet. Finding a workable solution to this issue is essential because, in the US, 95% of mobile communications are read and reacted to within three minutes of receipt [9]. To solve this issue, we propose a Natural Language Processing based strategy for precisely recognizing spam promotional SMS texts. In this paper, the issue of spam promotional SMS texts is thoroughly examined, along with several potential solutions. The dataset used consists of promotional and non-promotional SMS messages that were collected from our personal devices and online datasets. The data was preprocessed and converted into a format that is appropriate for machine learning models. Five alternative models, including Multinomial naive bayes and Support vector machines, then Logistic regression, Random forests, and finally a Multi-layer perceptron classifier, were used to categorize the SMS texts. Random Forest had the highest accuracy with 93.51%. Our proposed approach offers a potential solution to the increasing problem of spam promotional SMS messages, providing a means to filter out unwanted messages and improve the SMS messaging experience for users. By accurately distinguishing between promotional and non-promotional SMS messages, we aim to reduce the number of unwanted messages that users receive.

II. Literature Review

Spam SMS is a significant issue that impacts individuals all around the world, causing aggravation, frustration, and even security issues. In order to avoid this problem, professionals and researchers are doing tests to determine the best ways to filter and block unwanted text messages. From the study of research paper [1], with success rates of up to 98%, the Bayesian technique has been demonstrated to be quite efficient at removing spam SMS texts. This method makes use of statistical analysis to assess the likelihood that a specific message is spam based on particular keywords or other features. As you pointed out, there can be alternative strategies that are just as successful. One such strategy is the deterministic SQL query method. Based on variables including the reputation of the sender and the message's content, this method applies a preset set of criteria to identify spam communications. Various factors in the data being examined, such as the prevalence of particular spam message types or the frequency of false positives, may affect each approach's efficacy. The size and complexity of the data collection, the resources available for analysis, and the required level of accuracy

and precision will all affect which strategy is ultimately the most successful. In order to identify spam in textual data by extracting semantic information, it should be noted that the Multinomial Naive Bayes classification approach has been in [2] effectively applied to classify damaging Bangla text content at the sentence level with an accuracy percentage of 82.44%. This strategy can be useful for removing spam communications, especially in languages like Bangla where conventional filtering techniques might not be as efficient. It is crucial to remember that this method's efficacy might change based on the language utilized and the type of spam messages being examined. To evaluate its performance in other languages and further increase its accuracy, more study may be required, nevertheless, the preliminary findings are encouraging and imply that this strategy may be an effective weapon in the battle against spam communications. In the paper [3], the proposed model with Naive Bayes gave results that showed an accuracy of 98% in detecting spam for email. They have collected data from Kaggle and Sklearn. Moreover, to remove the noisy and unstructured data they followed a few natural language processing techniques including the removal of stop words, tokenization, and Bag of words (BOW). In paper [4], for detecting spam SMS, the LSTM model was used where the proposed model provided a recognition accuracy of 98.5%. They trained their model on 5574 rows of sample datasets where 4827 of these are determined to be legitimate while the remaining 747 are considered to be spam. In paper [5], To identify the text data and detect spam SMS, the authors used a variety of machine learning and deep learning methods.The researchers applied LSTM, a deep learning algorithm including multiple machine learning models, and achieved an accuracy of 98.5%. They trained their model on a dataset containing a total of 5572 rows with two columns. In the paper [6], the proposed Random Forest model fared best, with an accuracy rate of 93.60% for Bangla Spam Email Classification. For this research, the authors collected their own dataset. To gather at least 4500 in Bangla spam emails and about 800 in Bangla spam emails for the training dataset and the testing dataset respectively, they ran a survey. The usefulness of word embedding in categorizing spam emails is discussed in this work [7]. BERT, a pre-trained transformer model, is optimized to carry out the duty of separating spam emails from legitimate emails (HAM). To eliminate bias, two open-source data sets—2000 spam and 3000 ham—were combined. BiLSTM served as the foundational model. The BERT transformer model achieved the highest F1 score of 98.66% and the highest accuracy of 98.67%. The study in [8] combines the HHO algorithm with the k-NN algorithm to identify bogus spammers. The authors use the KNN (K-Nearest Neighbor) algorithm to categorize spam emails. However, when compared to the other algorithms, the accuracy of the Binary HHO approach was 94.3%, which was higher. Social media's dissemination of false medical information and rumors during the COVID-19 outbreak has left people perplexed and afraid. Prior studies have concentrated on applying machine learning and deep learning techniques to classify spam. These methods, however, have two significant drawbacks: First, human feature engineering is needed for machine learning models, which takes time and might not capture all important features. Second, deep neural networks may not be appropriate for real-time applications due to their high processing cost. The paper [9] proposes a deep ensemble model dynamic in nature for detecting spam that automatically extracts features and modifies its complexity in order to get around these drawbacks. Convolutional and pooling layers are used in the model to extract features, while random forests and highly randomized trees are used as basis classifiers to categorize texts as spam or real. In order to enhance classification performance, the model also uses ensemble learning techniques including boosting and bagging. The outcomes demonstrate the proposed model's efficacy in spam identification, with high precision, recall, f1-score, and accuracy of 98.38%. By using text preparation techniques and text classification algorithms to elucidate the true significance of viewpoints represented in text, researchers' workloads can be reduced. The goal of this study [10] was to find a link that works effectively with multilingual and cross-language data. Five different datasets from English, Bangla, and Banglish (transliterated datasets) were used in the experiment. Byte-mLS beat the XGB classification model with mLS preprocessing, attaining above 90% accuracy on Datasets in English, 93.43% accuracy on datasets in Bangla, and 78% accuracy on datasets that are transliterated. On English datasets, the mLSTM preprocessing method outperformed earlier approaches, and it did well overall.

III. Dataset

In order to develop and evaluate text classification models for spam detection in a multilingual context, we collected a diverse dataset of text messages that include both spam and ham messages. The dataset consists of a total of $2,876$ messages, which were obtained from a variety of sources. We used our own personal devices to collect a portion of the dataset, while the remainder was obtained from publicly available online datasets [12] [13] [14]. The messages in our dataset are written in Bangla, English, and Bangla-English mixed languages. This multilingual aspect of the dataset provides a unique challenge for spam detection, as the linguistic characteristics of spam messages can vary widely across different languages. Each message in the dataset is labeled as either spam or ham. There are a total of $1,417$ spam messages and $1,459$ ham messages, providing a roughly balanced dataset for training and evaluation. This balance is important for ensuring that models are not biased towards one class or the other. The dataset provides a representative sample of real-world text messages in a multilingual context, allowing for the development and evaluation of text classification models for spam detection in different languages.

IV. Classification and Results

In order to classify the SMS messages, five different models were implemented and after comparing the performances of all the models, it was found that Random Forest, Support

Vector Machine, and Multi-Layer Perceptron Classifier were the most accurate in classifying the messages into spam or ham categories.

DataPreprocessing: The 'sklearn' library has been used for label encoding the target variable. The 'torchtext' library has been used to create a vocabulary and tokenize the input text data. A 'BasicTokenizer' class was defined to perform tokenization, which includes splitting on punctuation and lowercasing the text. The tokenizer function is applied to all text messages, and the resulting tokens were used to build a vocabulary. Finally, the text messages were tokenized and converted into tensors, and then padded to create sequences of the same length. The processed data was then fed into the machine learning models for training and evaluation.

Models Implementation and Results: Using a train-test split with a test size of 30% and a random state of 42, the dataset was split into training and testing sets. To identify the ideal set of parameters, GridSearch was used to tweak the hyperparameters of the ml models. Multinomial Naive Bayes, Random Forest, Logistic Regression, Support Vector Machine, and Multi-Layer Perceptron Classifier were five of the machine learning models used. The accuracy of the Random Forest algorithm was the greatest at 93.51%. Accuracy values for the Support Vector Machine and Multi-Layer Perceptron Classifier were 86.79% and 85.63%, respectively. The accuracy for Logistic Regression was 76.25%, whereas the accuracy for Multinomial Naive Bayes was 74.62%.

Random Forest: Implementing Random Forest for the given dataset had an accuracy of 93.51%. The heat map of the confusion matrix and the learning curve for the Random forest algorithm is shown in Figure 1. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

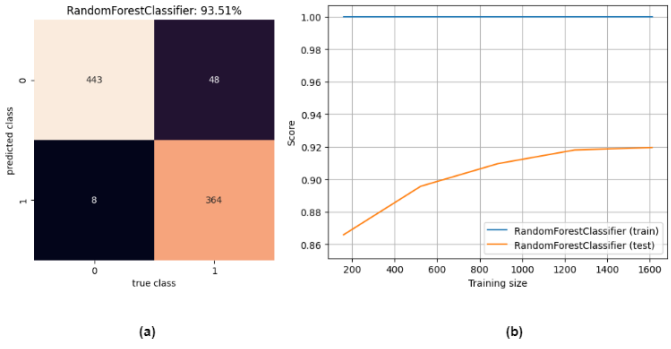

Figure 1: Confusion Matrix and Learning Curve of Random Forest

The following Table 1 summarizes the performance of the Random Forest algorithm through the classification report.

	class	precision	recall	f1score	support
	0	0.90	0.98	0.94	451
	1	0.98	0.88	0.93	412
Micro avg		0.94	0.93	0.93	863
Weighted avg		0.94	0.94	0.93	863

Table 1: Classification Report for Random Forest

Support Vector Machine: Implementing a Support Vector Machine for the given dataset had an accuracy of 86.79%. The heat map of the confusion matrix and the learning curve for the Support Vector Machine is shown in Figure 2. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

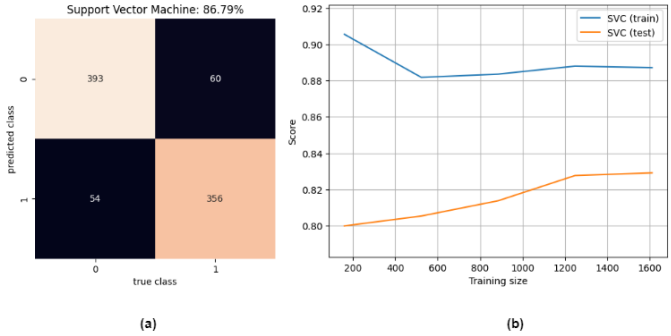

Figure 2: Confusion Matrix and Learning Curve of Support Vector Machine

The following Table 2 summarizes the performance of the Support Vector Machine through the classification report.

	class	precision	recall	f1score	support
	0	0.87	0.88	0.87	447
	1	0.87	0.86	0.86	416
Micro avg		0.87	0.87	0.87	863
Weighted avg		0.87	0.87	0.87	863

Table 2: Classification Report for Support Vector Machine

Multi-Layer Perceptron Classifier: Implementing Multi-Layer Perceptron Classifier for the given dataset had an accuracy of 85.63%. The heat map of the confusion matrix and the learning curve for the Multi-Layer Perceptron Classifier is shown in Figure 3. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

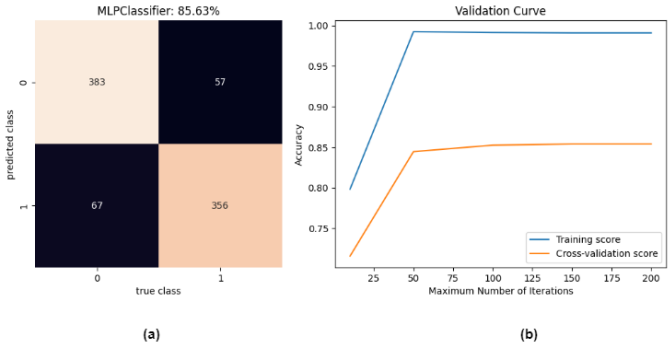

Figure 3: Confusion Matrix and Validation Curve of MLPClassifier

The following Table 3 summarizes the performance of the MLPClassifier through the classification report.

	class	precision	recall	f1score	support
	0	0.87	0.85	0.86	450
	1	0.84	0.86	0.85	413
Micro avg		0.86	0.86	0.86	863
Weighted avg		0.86	0.86	0.86	863

Table 3: Classification Report for MLPClassifier

LogisticRegression: Implementing LogisticRegression for the given dataset had an accuracy of 76.25%. The heat map of the confusion matrix and the learning curve for the LogisticRegression is shown in Figure 4. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

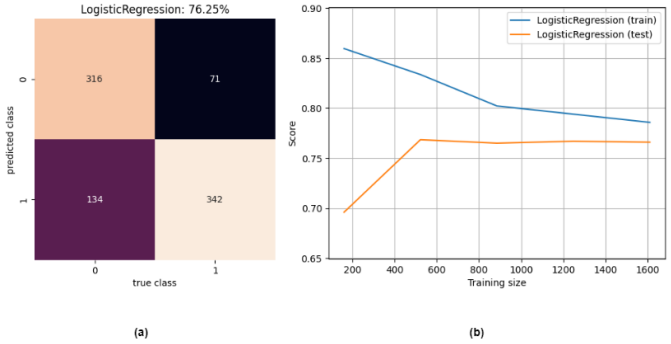

Figure 4: Confusion Matrix and Learning Curve of LogisticRegression

The following Table 4 summarizes the performance of the LogisticRegression through the classification report.

	class	precision	recall	f1score	support
	0	0.82	0.70	0.76	450
	1	0.72	0.83	0.77	413
Micro avg		0.77	0.77	0.76	863
Weighted avg		0.77	0.76	0.76	863

Table 4: Classification Report for LogisticRegression

Multinomial Naive Bayes: Implementing Multinomial Naive Bayes for the given dataset had an accuracy of 74.62%. The heat map of the confusion matrix and the learning curve for the Multinomial Naive Bayes is shown in Figure 5. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

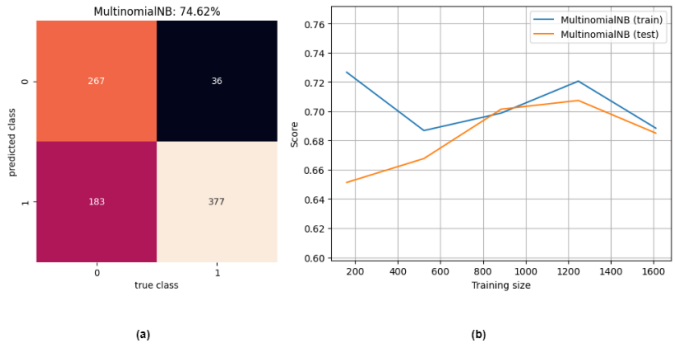

Figure 5: Confusion Matrix and Learning Curve of MultinomialNB

The following Table 5 summarizes the performance of the Multinomial Naive Bayes through the classification report.

	class	precision	recall	f1score	support
	0	0.88	0.59	0.71	450
	1	0.67	0.91	0.77	413
Micro avg		0.78	0.75	0.74	863
Weighted avg		0.78	0.75	0.74	863

Table 5: Classification Report for MultinomialNB

Model Comparison and Analysis: The following bar chart presented in Figure 6 depicts a comparison of model accuracies. Among the models, Random Forest had the highest accuracy rate of 93.51%. Support Vector Machine and Multi-Layer Perceptron Classifier had accuracies of 86.79% and 85.63%, respectively. Logistic Regression recorded an accuracy rate of 76.25%, while Multinomial Naive Bayes had the lowest accuracy rate of 74.62%. The results indicate that the Random Forest model performed the best among the models compared in terms of accuracy.

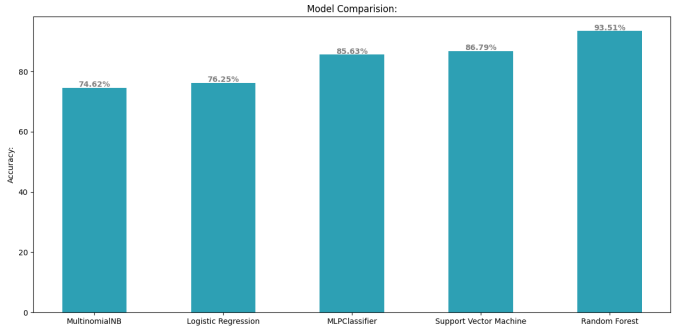

Figure 6: Model accuracy Comparison

Calibration Curve: Figure 7 presents a calibration curve that compares the predicted probabilities of the models to the actual outcomes. The diagonal line represents an ideal calibration curve, and the closer the plotted lines are to it, the better calibrated the model is. Based on the figure, Random Forest is the best-calibrated model as it is closest to the diagonal line. Support Vector Machine is also well calibrated, while MLPClassifier, Logistic Regression, and MultinomialNB are the least well calibrated, with a significant deviation from the diagonal line. The calibration curve analysis suggests that Random Forest is the most reliable model among the compared models.

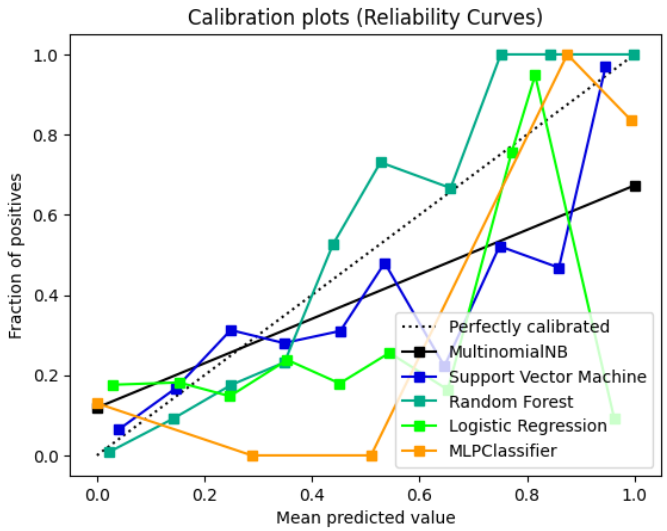

Figure 7: Calibration Curve

V. Future Work and Conclusion

A. Future Work:

To further improve this study's findings, future work should focus on increasing the dataset size by collecting more real-world data. This will enable the use of more complex architectures, transformers, and models to achieve better accuracy. Additionally, the classification of spam into different sub-parts can be explored, along with the classification of different types of spam texts. Such efforts will further enhance the effectiveness and efficiency of spam filtering systems, making them more reliable and robust. Furthermore, the dataset can be expanded to include more languages to enable more diverse classifications. This would help to improve the adaptability of the system to different regions and users.

B. Conclusion:

In conclusion, the problem of spam promotional SMS messages has become increasingly prevalent in recent years, creating a frustrating and intrusive experience for users. While SMS messaging offers businesses a direct means of communication with customers, it is also vulnerable to spam and phishing assaults. In this research paper, we presented a machine learning and deep learning-based approach for accurately detecting spam promotional SMS messages. By collecting our own dataset of multilingual SMS messages and employing several machine learning models, including Random Forest, Support Vector Machine, and Multi-Layer Perceptron Classifier, we were able to achieve a high accuracy of up to 93.51% in detecting spam messages. Our proposed approach provides a potential solution to the problem of unwanted SMS messages, allowing for the filtering out of unwanted messages and improving the overall messaging experience for users. The multilingual aspect of our dataset provides a unique challenge for spam detection, and our approach can be applied to various languages, making it a valuable contribution to the field of text classification for spam detection.

References

[1] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, China, 2011, pp. 101-105, doi: 10.1109/ICCSNT.2011.6181918.

[2] T. Islam, S. Latif and N. Ahmed, "Using Social Networks to Detect Malicious Bangla Text Content," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-4, doi: 10.1109/ICASERT.2019.8934841.

[3] Abid, Muhammad Ullah, Dr. Saleem Siddique, Muhammad Mushtaq, Muhammad Aljedaani, Wajdi Rustam, Furqan. (2022). Spam SMS filtering based on text features and supervised machine learning techniques. Multimedia Tools and Applications. 81. 10.1007/s11042-022-12991-0.

[4] Nyamathulla, et al. "SMS Spam Detection With Deep Learning Model." Journal of Positive School Psychology, vol. Vol. 6, no. No. 5, 2022, pp. 7006–13. journalppw.com.

[5] S. Gadde, A. Lakshmanarao and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 358-362, doi: 10.1109/ICACCS51430.2021.9441783.

[6] R. Amin, M. M. Rahman and N. Hossain, "A Bangla Spam Email Detection and Datasets Creation Approach based on Machine Learning Algorithms," 2019 3rd International Conference on Electrical, Computer Telecommunication Engineering (ICECTE), Rajshahi, Bangladesh, 2019, pp. 169-172, doi: 10.1109/ICECTE48615.2019.9303525.

[7] Shanafelt TD, Dyrbye LN, West CP, Sinsky C, Tutty M, Carlasare LE, Wang H, Trockel M. Suicidal Ideation and Attitudes Regarding Help Seeking in US Physicians Relative to the US Working Population. Mayo Clin Proc. 2021 Aug;96(8):2067-2080. doi: 10.1016/j.mayocp.2021.01.033. Epub 2021 Jul 20. PMID: 34301399.

[8] Al-Betar, Mohammed Mashaleh, Ashraf Yaseen, Qussai Mustafa, Hossam Ibrahim, Noor. (2022). Detecting Spam Email with Machine Learning Optimized with Harris Hawks optimizer (HHO) Algorithm. 10.1016/j.procs.2022.03.087.

[9] Shaaban, Mai A., et al. "Deep Convolutional Forest: A Dynamic Deep Ensemble Approach for Spam Detection in Text." Complex Intelligent Systems, vol. 8, no. 6, Springer Science and Business Media LLC, Apr. 2022, pp. 4897–909. Crossref, https://doi.org/10.1007/s40747-022-00741-6.

[10] T. Khan, D. D. Mallick, M. S. I. Khan, M. M. Hasan and F. B. Ashraf, "An Efficient Text Preprocessing and Classification Technique for Multilingual and Transliterated Data," 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2022, pp. 366-371, doi: 10.1109/ICCIT57492.2022.10054834.

[11] Mannheimer, Simon. "USA Text Message Statistics Updated for 2023." SMS Comparison, 16 Apr. 2023, www.smscomparison.com/sms-statistics.

[12] mishra, sandhya; Soni, Devpriya (2022), “SMS PHISHING DATASET FOR MACHINE LEARNING AND PATTERN RECOGNITION”, Mendeley Data, V1, doi: 10.17632/f45bkkt8pr.1
[13] Ahmed, Md Faisal; Mahmud, Zalish; Biash, Zarin Tasnim ; Ryen, Ahmed Ann Noor ; Hossain, Arman ; Ashraf, Faisal Bin (2021), “Bangla Online Comments Dataset”, Mendeley Data, V1, doi: 10.17632/9xjx8twk8p.1
[14] Rahman, Md Ataur; Seddiqui, Md. Hanif (2020), “BanglaEmotion: A Benchmark Dataset for Bangla Textual Emotion Analysis”, Mendeley Data, V1, doi: 10.17632/24xd7w7dhp.1