Bangla Documents Classification using Transformer Based Deep Learning Models

Md Mahbubur Rahman*, Md. Aktaruzzaman Pramanik†, Rifat Sadik†, Monikrishna Roy‡, Partha Chakraborty§
*Crowd Realty, Tokyo, Japan, mahbuburrahman2111@gmail.com
†Dept. of CSE, Jahangirnagar University
{a.pramanikk, rifat.sadik.rs}@gmail.com
‡Samsung R&D Institute, Dhaka, Bangladesh, mkroy.cs@gmail.com
§Dept. of CSE, Comilla University, Cumilla, Bangladesh, partha.chak@cou.ac.bd

2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI) |978-1-6654-0489-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/STI50764.2020.9350394

***Abstract*—Document classification or categorization assign documents to a predefined domain category. The improvement of document classification techniques has been noticeable worldwide recently. Many transformer-based models have been introduced for different languages, which shows significant improvement in this area of Natural Language Processing. In this paper, we have classified Bangla text documents with the most recent transformer or attention mechanism-based models. We have applied the BERT (Bidirectional Encoder Representations from Transformers) and ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) model for Bangla text classification. Both of them are pre-training text encoders and we have applied fine-tuning approach for the downstream(classification) task. Here, we have used three different Bangla text datasets for our experiment. Both of the models provide outstanding performance for two out of three datasets we have used.**

***Keywords*—NLP, Deep Learning, Transformers, Bangla Text Classification, BERT, ELECTRA**

I. Introduction

Document classification or Text Categorization is a very essential tool for retrieving necessary information from a vast amount of data with reduced cost and time. Several techniques and algorithms have been applied over the years for this purpose [1], and the process of improving those techniques are still carrying out significantly. BERT (Bidirectional Encoder Representations from Transformers) is one of the most used transformers used for document or text classification.

Plenty of researches have been done recently using BERT-based classification [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. In [6] & [11], they have performed sentence categorization using BERT based transformers but [6] used a pre-trained model and fine-tuning method which has given far better accuracy for their sequential sentence classification. While [2], [5] and [9] also used BERT model where they applied fine tuning in their experiment. Other BERT based transformers were used either adding extra polling layers [4] or using BERT based hybrid model [10]. So, in our work we applied BERT model using both pre-training and fine-tuning approach to classify Bangla documents.

While in the case of Bangla text or document classification, some researchers applied the SGD classifier [13], in some works, many popular algorithms such as Decision Tree, KNN, SVM were applied [14]. In [15], they have applied SVM classifier reformed with TF-IDF method.

We know the multilingual BERT provides pre-trained language models for more than 100 languages and by fine-tuning, we can make it very useful for Bangla text or document classification. Whereas ELECTRA [16] is one of the most recent improvements in transformer-based NLP tasks. Hence in this paper, we have used BERT and ELECTRA models, and the accuracy for ProthomAlo and BARD dataset using these two models was very satisfactory in the arena of Bangla document classification. Our approach can be used for building more robust and automated NLP applications, which will create a significant impact in "Sustainable Technologies for Industries 4.0".

We have used three different datasets having a different number of classes and have shown their efficacy with two latest transformer models using transfer learning technique, hence our contribution in this paper-

- We have applied BERT and ELECTRA transformers for Bangla Document classification, which provide satisfactory results.
- We have used three different datasets with several numbers of classes that are used in performance evaluation of the models.
- We have shown the performances of the models used for different datasets using several performance metrics

The remaining part of this paper is organized as follows: Related works have been presented in section II. Section III describes the methodology shortly. Section IV describes details about dataset collection, processing, and also the details about the models used to train the dataset. Section V discusses the performance metrics, evaluation of the model results, and findings of this experiment. Finally, the conclusion has been drawn with future directions in section VI.

II. Literature Review

Bangla document or sentence categorization is an emerging field of research with an increase in the availability of web-based documents. Fasihul Kabir et al. [13] used Stochastic

978-1-6654-0489-1/20/$31.00©20XX IEEE

Gradient Descent (SGD) as a classifier to categorize Bangla text. Feature extraction involved was done using the Term Frequency Inverse Document Frequency (TF-IDF) method. This method achieved higher recall and f1 score (0.9388% and 0.9385%) when compared to other classification algorithms like Naive Bayes and SVM. Supervised algorithms namely Decision Tree, KNN, Naïve Bays, and SVM were used by Ashis Kumar Mandal and Rikta Sen [14] for the Bangla document classification task. Different Bangla web sources were used to evaluate these models. The SVM classifier outperforms others by achieving a higher F1 measure of 89.14. Md Saiful Islam et al. [15] proposed an approach where SVM with TF-IDF was to classify the Bangla document. This method was used to classify documents of 12 categories from a Bangla corpus and achieved an accuracy of 92.52%. Document classification by measuring distance was proposed by Ankita Dhar et al. [17], where Cosine and Euclidean distance measures were used in this case. The cosine distance measure gained the highest accuracy, followed by Euclidean distance measures in classifying documents from different domains such as medical, business, sports, etc.

For sentiment analysis Chi Sun, Luyao Huang, Xipeng Qiu [2] proposed a method that first constructed auxiliary sentences then classified them into their respective sentence pairs, for improvement fine-tuning was done for the BERT model. The latter comparison was analyzed between single sentence and pair sentence classification tasks. The experiment was conducted on the SentiHood dataset and proposed BERT-pair QA-B and NLI-B achieves higher accuracy and AUC values. Adhikari et al. [3] proposed a refined model of BERT that transfers knowledge from the base model to a smaller variant that used 30x fewer parameters. The experiment was conducted on four different datasets. The result showed that the proposed knowledge transfer strategy achieved a higher F1-score for all datasets compared to other traditional models. In sentiment analysis, Youwei Song et al. [4] modified the BERT model by adding extra polling layers that follow different polling strategies. This type of implementation fine-tuned the model. The modified model was experimented on two different datasets, namely ABSA datasets and SNLI datasets, both cases achieved boosted results compared to the traditional models. Xin Li et al. [5] proposed the BERT model for the E2E-ABSA task by building some baselines and fine-tuning the model. Datasets from the SemEval repository were used to train and test the proposed model. The highest F1 score was achieved by combining BERT with GRU and SAN model (61.12, 74.72). After fine-tuning the models, improvements had been seen, and BERT with the TFM model achieved an F1 score of 74.41.

To sentence categorization, Cohan et al. [6] presented a model based on BERT, which is a pre-trained model. The sentences were contextually analyzed based on the forming word and this word representation was modeled by the transformer layers. The highest F1 score for the proposed model was achieved for the PUBMED dataset was 92.9, which outperformed other models. To classify books based on the descriptive texts Ostendroff et al. [7] combines the BERT model with knowledge graph embedding for WIkidata and other metadata. Besides, the author's information was used in this classification task. The proposed model achieved an F1 score of 87.20 for tasks that comprise eight labels. Pavlopoulos et al. [8] used the BERT model to identify abusive language on social media and categorize them. The first baseline of this approach used the perspective API to detect offensive language, and the second approach comprised of BERT that was used to categorize these offensive languages. For evaluation, a dataset named SEMEVAL-2019 OFFENSEVAL was used containing tweets. The prospective baseline achieved an accuracy of 83%, which was higher than the BERT baseline.

To extract relation and entity from the medical text, Kui Xu et al. [9] proposed a model that utilizes BERT with fine-tuning. Here attention model was combined with attention mechanisms for better feature representation. Layers of the BERT model was changed using STR-encoder for using the prior knowledge of the model. For entity recognition, the model gained an F1 score of 96.89%, and for relation classification, it achieved an 88.51% F1 score. Shanshan Yu et al. [10] enhanced the performance of the conventional BERT model using a hybrid model called BERT4TC. For better results, auxiliary sentences were constructed and then converting them into sentence pairs while preprocessing. The experiment was conducted upon several datasets and among which the proposed model achieved higher accuracy of 0.9987% for DBPedia. Zhengjie Gao et al.[11] proposed a BERT model for sentence classification that worked on aspect level, and the effect of replacement of embedding transformation with BERT had been observed. The experiment was conducted upon three independent datasets and achieved 70% to 80% accuracy. For NER (Named Entity Recognition) task, Imranul Ashrafi et al. [12] used Word2Vec with the BERT model. In this task, loss function was measured, which was used to learn about the penalty in the learning process. An additional experiment was conducted by adding the CRF layer. 1st scenario in which BERT and BiLSTM were used achieved a micro F1 score of 90.65 and MUC of 71.04. When the CRF layer added with BERT and BiLSTM, and CW model, the MUC score was 72.04.

III. Methodology

At first, we collected data from three open-source data repositories. We pre-processed the data to make ready for the transformer-based model. After that, we set the BERT model with a pre-trained model and fine-tuned the model with our collected data. Finally, we evaluated the model based on different metrics. Figure 1. describes the process briefly.

IV. Experimental Details

A. Data Collection

We collected three different open-source data: BARD[18], OSBC[1] and ProthomAlo[2]. BARD, OSBC, and ProthomAlo have respectively 5, 11, and 6 unique classes.

[1]https://scdnlab.com/corpus/

[2]https://www.kaggle.com/twintyone/prothomalo

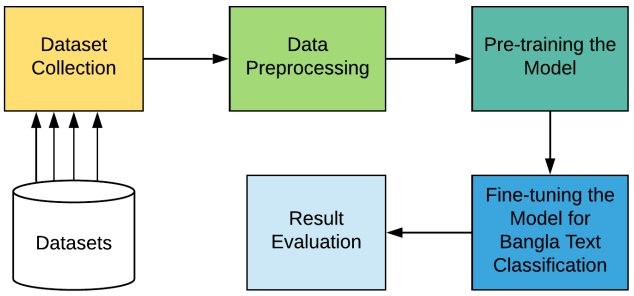

Fig. 1. Proposed System Architecture(A schematic representation of the methodology including data collection, pre-processing, training and evaluation)

B. Data Pre-processing

We considered only the Bangla language. So if any text has more than 20% characters from outside of the Bangla language, we removed these texts from the data. We selected 50560 data from the BARD, 78796 data from the OSBC, and 128761 data from the ProthomAlo dataset. We removed some unnecessary punctuations. Finally, we divided the data into train and test sets with an 80:20 ratio. Table I shows a brief description of each class of three datasets.

TABLE I
EXPLANATION OF THREE DIFFERENT DATASETS.

	BARD		OSBC		ProthomAlo	
Labels	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
sports	10322	2581	8760	2190	35270	8817
international	5505	1376	4511	1128	23432	5853
entertainment	3726	932	6882	1720	21538	5385
economy	4894	1224	4096	1024	12914	3229
technology	-	-	1742	436	7222	1805
crime	-	-	5719	1430	-	-
environment	-	-	5133	1283	-	-
opinion	-	-	4719	1180	-	-
art	-	-	1720	430	-	-
politics	-	-	14860	3715	-	-
accident	-	-	6882	1720	-	-
education	-	-	-	-	2647	662
state	16000	4000	-	-	-	-

C. Training

We used two different transformer-based models: BERT, ELECTRA. These models' training generally has two parts. The first part is pre-training and the second part is fine-tuning. The pre-training part is unsupervised learning and is trained on a large set of an unlabeled corpus for language modeling. The fine-tuning part is done by initializing the same parameter of the pre-training task and adding a fully connected layer based on the downstream task. We didn't perform the pre-training part in this task and we only performed the fine-tuning part. We collected the model weights from huggingface[3]. The weights[4] which was used with BERT was trained by the data of 102 languages and the weights[5] which was collected for ELECTRA was trained with only Bangla corpus. We used 20 epochs and batch size 16 to fine-tune the weights. We saved the best weights based on the validation loss during the fine-tuning.

1) M-BERT: The BERT model architecture consists of an encoder of 12 transformer blocks, 12 self-attention heads and 768 hidden states.

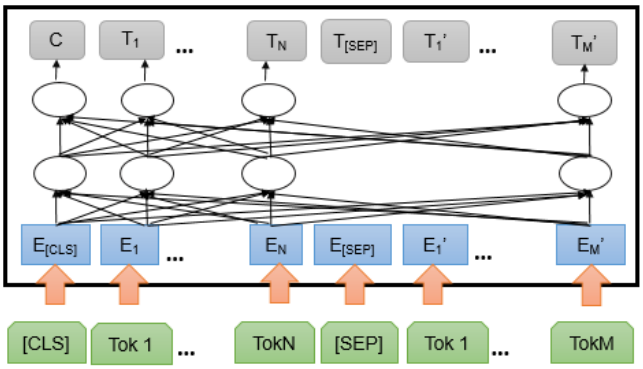

Fig. 2. Input and output pattern of a BERT based model. Same architecture used in all kinds of downstream tasks but based on the task, we need to add the final fully connected layer or classifier.

The pre-training task of BERT [19] depends on two unsupervised sub-tasks: masked language modeling (MLM) and next sentence prediction(NSP). These two sub-tasks use the same model architecture but with different input patterns and different output layer. In MLM, a fixed amount of tokens of the input sequences is masked and the model is trained for predicting the original tokens of the masked tokens. In NSP, the model has to predict whether two sequences of text are naturally following each other or not. 50% data is generated automatically by taking sentence pairs next to each other and the other 50% is generated by taking sentence pairs randomly from the unlabeled corpus.

Fig. 2 shows overall procedure for pre-training and fine-tuning but without output layer. In MLM, a special classification token[cls] is added before each input sequences and a separator token[SEP] is used after each input sequences(3 . But in case of NSP, a separator token([SEP])(Fig. 4) is also used between the pair of sentences of each input sequences to separate them as each input sequence of NSP is a combination of two sentences. The initial input embedding(E_{Tok}) is calculated by summing up the token, sentence and positional embedding.

In the case of MLM, the final hidden vector of each of the masked tokens is passed to a softmax classifier(output layer) to predict the original token.

On the other hand, during NSP, the final hidden vector(C) of the [CLS] token is fed to a binary classifier(output layer) to predict whether the input pair is following each other or not.

In the fine-tuning part for downstream tasks, the final hidden vector(C) of the first token[CLS] is sent to a fully connected

[3]https://huggingface.co/

[4]https://huggingface.co/bert-base-multilingual-uncased

[5]https://huggingface.co/monsoon-nlp/bangla-electra

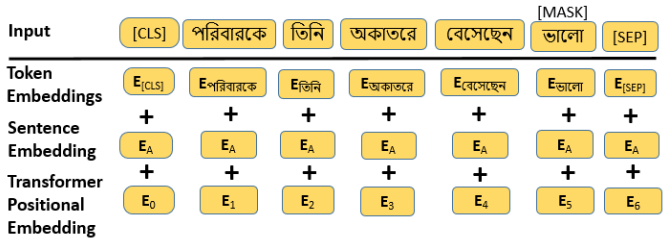

Fig. 3. Input pattern for MLM subtask. A special classification token[CLS] is used before each input sequence and a separator token[SEP] is used after each input sequence. The initial input embedding is calculated by summing up the token, sentence and positional embedding.

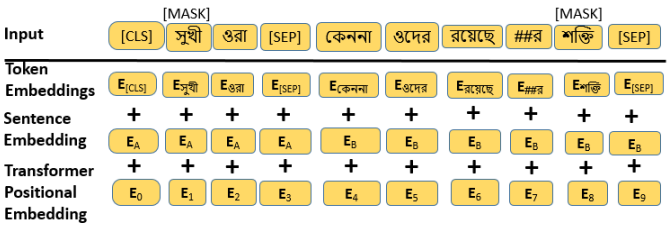

Fig. 4. Input pattern for NSP subtask. A [CLS] is used before each input sequence and a separator token [SEP] is used between each sentence of an input sequence. The initial input embedding is calculated by summing up the token, sentence and positional embedding.

layer to classify the text. Finally, parameters of the model are fine-tuned by the labeled data.

2) ELECTRA: While MLM pre-training process in BERT is applied by replacing some tokens with MASK and then try to reconstruct the original tokens by training a model that requires a huge amount of computations. ELECTRA aims to deploy less compute resources for pre-training than BERT. ELECTRA primarily consists of two neural network a generator(MLM) and a discriminator(ELECTRA) (Fig. 5). Each of them generally has an encoder. The generator is trained to perform masked language modeling. It randomly selects some input token and masks them by [MASK] token. The generator is then trained to predict the original token of the masked form. The discriminator then tries to predict if each token was replaced by the generator or not.

During the fine-tuning part on downstream tasks, the generator is dropped, a fully connected layer is added after discriminator and finally, the discriminator is fine-tuned by the labeled sample data.

D. Testing

After training, we loaded the best weights saved during the training. After that we test the models with the test data.

V. Result Analysis

A. Performance Evaluation

How well a deep learning model performs can be inferred from some specific parameters. Precision, Recall, F1 score and Accuracy are the foremost parameters [20] that are used to measure the performance of a deep learning model. The values

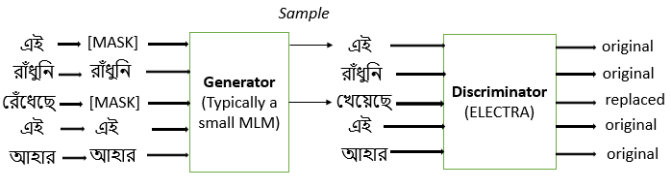

Fig. 5. Procedure of pre-training using ELECTRA. For fine-tuning, we need to drop generator and add a fully connected network after discriminator.

of these measures are calculated during the training and testing process. The formulas [21] that are used in this computations are given in equation 1, 2, 3, 4.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Where TP, FP, TN, FN are the true positive, false positive, true negative, and false negative respectively. The computed precision, recall, F1 score and accuracy for this study is presented in Table II, III, IV, V

TABLE II
AVERAGE PRECISION VALUES OF THE TWO MODELS FOR THE THREE DATASETS.

Models	ProthomAlo	BARD	OSBC
BERT	94.46%	90.40%	76.34%
ELECTRA	94.36%	92.01%	76.05%

TABLE III
AVERAGE RECALL VALUES OF THE TWO MODELS FOR THE THREE DATASETS.

Models	ProthomAlo	BARD	OSBC
BERT	93.85%	91.22%	70.85%
ELECTRA	94.02%	91.31%	73.48%

TABLE IV
AVERAGE F1 SCORES OF THE TWO MODELS FOR THE THREE DATASETS.

Models	ProthomAlo	BARD	OSBC
BERT	94.14%	90.83%	72.27%
ELECTRA	94.18%	91.60%	73.68%

TABLE V
ACCURACY OF THE TWO MODELS FOR THE THREE DATASETS.

Models	ProthomAlo	BARD	OSBC
BERT	96.09%	92.66%	76.90%
ELECTRA	96.39%	93.05%	78.53%

B. Discussion and Findings

From table II, a comparison based on precision is presented. It can be seen that, BERT outperforms ELECTRA for all three datasets. The highest Recall value (table III) was achieved for Prothom Alo dataset (94.36%) for ELECTRA. When it comes to comparing Recall values, ELECTRA performed well than BERT for all cases. By analyzing F1 score and accuracy from table IV and V, it can be observed that ELECTRA model achieved higher values (94.18% and 96.39%) for most of the cases. In case of OSBC dataset, we got lesser accuracy for both models because the dataset is not balanced enough containing many classes with different size of data.

In the case of BERT, the performance based on the given parameters was not prominent compared to ELECTRA. Masked Language Modeling or MLM is used in BERT which is responsible for limiting the tokens by replacing tokens with masks. This resulted in a low understanding of the language, that was used to train the model. Besides, there is a difference between token distributions which is another reason for low performance.

ELECTRA model achieved higher accuracy and the F1 score implies that it performs better than the BERT model. In ELECTRA, the token detection task is replaced and couples the generator with a discriminator. This allows us to predict the source of the tokens and identifies the original input sequence. This creates a higher efficiency for modeling and faster learning. ELECTRA masks out the input sequence unlike BERT and creates a bi-directional representation over the entire input sequence. Instead of defining tasks over all the sequences, ELECTRA replaces the final embedding layer and applies loss to every token in the sequence. For these reasons, ELECTRA provides a powerful modeling scheme and outperforms the BERT model.

VI. Conclusions

In this paper, a transformer based Bangla document classification was implemented. Two most recent transformer based models, namely BERT and ELECTRA, were used for this classification task. We measure the performance of our task based on accuracy, f1 score, recall, and precision. It has been inferred from our experiment that the ELECTRA model gained higher accuracy and f1 score while classifying different domains of Bangla documents of different data sources. The replacement of the token detection task with the creation of a discriminator and distribution of loss to every token gives advantages to the ELECTRA model over BERT to perform well.

Despite the fact that we have obtained much satisfactory result, our work has a future scope of improvement. Due to the lack of data and hardware resources, we could not use a well pre-trained model. So in the future, we would like to pre-train and fine-tune more transformer based models. Besides, experiments can be conducted using transformer models such as RoBERTa, ALBERT, CTRL, FlauBERT etc. as a part of future research.

References

[1] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[2] C. Sun, L. Huang, and X. Qiu, "Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence," *arXiv preprint arXiv:1903.09588*, 2019.

[3] A. Adhikari, A. Ram, R. Tang, and J. Lin, "Docbert: Bert for document classification," *arXiv preprint arXiv:1904.08398*, 2019.

[4] Y. Song, J. Wang, Z. Liang, Z. Liu, and T. Jiang, "Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference," *arXiv preprint arXiv:2002.04815*, 2020.

[5] X. Li, L. Bing, W. Zhang, and W. Lam, "Exploiting bert for end-to-end aspect-based sentiment analysis," *arXiv preprint arXiv:1910.00883*, 2019.

[6] A. Cohan, I. Beltagy, D. King, B. Dalvi, and D. S. Weld, "Pretrained language models for sequential sentence classification," *arXiv preprint arXiv:1909.04054*, 2019.

[7] M. Ostendorff, P. Bourgonje, M. Berger, J. Moreno-Schneider, G. Rehm, and B. Gipp, "Enriching bert with knowledge graph embeddings for document classification," *arXiv preprint arXiv:1909.08402*, 2019.

[8] J. Pavlopoulos, N. Thain, L. Dixon, and I. Androutsopoulos, "Convai at semeval-2019 task 6: Offensive language identification and categorization with perspective and bert," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 571–576, 2019.

[9] K. Xue, Y. Zhou, Z. Ma, T. Ruan, H. Zhang, and P. He, "Fine-tuning bert for joint entity and relation extraction in chinese medical text," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 892–897, IEEE, 2019.

[10] S. Yu, J. Su, and D. Luo, "Improving bert-based text classification with auxiliary sentence and domain knowledge," *IEEE Access*, vol. 7, pp. 176600–176612, 2019.

[11] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with bert," *IEEE Access*, vol. 7, pp. 154290–154299, 2019.

[12] I. Ashrafi, M. Mohammad, A. S. Mauree, G. M. A. Nijhum, R. Karim, N. Mohammed, and S. Momen, "Banner: A cost-sensitive contextualized model for bangla named entity recognition," *IEEE Access*, vol. 8, pp. 58206–58226, 2020.

[13] F. Kabir, S. Siddique, M. R. A. Kotwal, and M. N. Huda, "Bangla text document categorization using stochastic gradient descent (sgd) classifier," in *2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, pp. 1–4, IEEE, 2015.

[14] A. K. Mandal and R. Sen, "Supervised learning methods for bangla web document categorization," *arXiv preprint arXiv:1410.2045*, 2014.

[15] M. S. Islam, F. E. M. Jubayer, and S. I. Ahmed, "A support vector machine mixed with tf-idf algorithm to categorize bengali document," in *2017 international conference on electrical, computer and communication engineering (ECCE)*, pp. 191–196, IEEE, 2017.

[16] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[17] A. Dhar, N. Dash, and K. Roy, "Classification of text documents through distance measurement: An experiment with multi-domain bangla text documents," in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*, pp. 1–6, IEEE, 2017.

[18] M. T. Alam and M. M. Islam, "Bard: Bangla article classification using a new comprehensive dataset," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–5, IEEE, 2018.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[20] M. Junker, R. Hoch, and A. Dengel, "On the evaluation of document analysis components by recall, precision, and accuracy," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, pp. 713–716, IEEE, 1999.

[21] M. Zaman and C.-H. Lung, "Evaluation of machine learning techniques for network intrusion detection," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5, IEEE, 2018.