BiLSTM-Based Smishing Detection for Bangla SMS

Anmay Paul Arpan[1], Rajoshree Ghatak[1], Md.Mahmudul Hasan[1], Anuj Roy[1], Md Azijul Haque[1], and Sadman Sadik Khan[1*]

[1]Daffodil International University, Dhaka, Bangladesh

{arpan2305101696, ghatak2305101162, hasan23051715, anuj2305101913, azijul2305101765}@diu.edu.bd, sadman15-13696@diu.edu.bd*

Abstract. A morphologically sophisticated and diglossic Bangla is a difficult language for Natural Language Processing (NLP), particularly for security tools such as smishing (SMS-based phishing) detection. This paper proposes a Bidirectional Long Short-Term Memory (BiLSTM)-based model to identify Bangla SMS as normal, promotional, or smishing based on an evenly divided dataset of 2,772 messages. After preprocessing with tokenization, normalization, and padding, the model was trained with the Adam optimizer, class-weighted loss, and early stopping. Based on experimental outcomes, the BiLSTM achieved an overall accuracy of 95recall, and F1-score were averaged at 0.95. While normal and promotional SMS were put into the good performance class (F1 = 0.95 and 0.98, respectively), smishing messages attained a precision of 0.98 but recall of 0.89 which was lower due to misclassifications to the normal class. ROC analysis also confirmed strength with 1.00 AUC readings for normal and promotional, and 0.99 for smishing, establishing the benchmark of Bangla smishing detection and indicating the need for advanced techniques to reduce false negatives even further.

Keywords: Bangla SMS classification, Natural Language Processing (NLP), Bidirectional Long Short- Term Memory (BiLSTM), Smishing detection, SMSbased phishing, Morphologically rich languages, Lowresource language processing.

1 Introduction

Natural Language Processing (NLP) is one of the most groundbreaking domains of artificial intelligence that enables machines to understand, process, and classify human languages in applications that are applied in the real world. Short message classification has been a central element of spam blocking as well as phishing identification, as demonstrated by Aliza et al. [1]. SMS spam detection reviews confirm the central position of machine learning and NLP techniques in the field [2]. Recent comparative research has shown that deep learning and classical approaches are both effective, e.g., CNNs, RNNs, BiLSTMs, GRUs, and attention mechanisms [3]. Deep learning models, as proposed by Abayomi-Alli

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_35

et al., have also highlighted their capability of extracting SMS contextual patterns [5]. At the same time, systematic reviews identified the persisting challenge of spam detection in the digital and mobile environment [4]. For Bangla, spam SMS detection is a rapidly emerging area of study. Uddin et al. [10] have used transformerbased models on Bangla with successful results. Aliza et al. have also explored using BERT and ELMo in building Bangla SMS datasets for classification [11]. Hybrid models for Bangla smishing SMS detection have recently been proposed by Tanbhir et al. [12]. There should be strong Bangla datasets. Johari et al. offered valuable remarks on proposed datasets for spam detection [9], and their BangalaBarta dataset was an invaluable resource for the detection of spam as well as smishing SMS [13]. Balanced datasets like these ensure more accurate testing and reduce bias while training. Here, we have trained a BiLSTM model with a balanced Bangla SMS dataset of 924 samples per class. It has been proven in the previous studies that BiLSTMs are effective for SMS spam classification, as shown by Aparna et al. [14]. Our model achieved an overall accuracy of 95% with high precision and recall values consistently across classes. The identical results on deep learning performance on spam detection were observed by Altunay et al. while testing Turkish and English SMS [6]. The confusion matrix shows that most of the errors happen when smishing messages are wrongly classified as normal. This finding aligns with the results of De Goma et al., where deep learning struggled to differentiate between phishing-like and genuine texts [7]. The same issues were reported by Shinde et al. in OCR-based SMS scam detection [8]. Performance evaluation with learning curves and ROC analysis validates the model's robustness. The high AUC scores, similar to what was achieved by Abayomi-Alli et al., reflect the discriminatory power of BiLSTM models [5]. However, the relatively lower recall for smishing points out scope for improvement. For example, Airlangga's comparative study pointed out that longer sequence modeling can improve classification [15]. Similarly, Johari et al. demonstrated that ensemble models with LLMs can reduce false negatives in spam detection [16]. The ability to classify Bangla SMS into normal, promotional, and smishing has significant practical uses. It enhances spam filtering in cellular networks, as noted by the Telecom Regulatory Authority of India in their spam control guidelines [21]. It also enhances NLP research for low-resource languages, as evident from recent Bangla-focused research [12].

2 Related Work

Spam and phishing detection via SMS has been a very active area of research in natural language processing (NLP) and machine learning (ML), with diverse approaches ranging from simple classifiers to recent deep learning and transformer-based models. Past studies have employed ML-based techniques such as Naïve Bayes, SVM, and decision trees with moderate success at detecting unwanted SMS but stumbling with contextual nuances and adversarially trained spam [1], [2], [4]. To overcome these shortcomings, hybrid architecture and neural models emerged to the forefront. Abayomi-Alli et al., for instance, employed a deep

learning pipeline for SMS spam filtering with utmost improvement in comparison to conventional ML [5], while Airlangga compared some neural architectures like CNN, LSTM, BiLSTM, GRU, and attention mechanisms and determined that BiLSTM ranked top for sequencebased text categorization [3], [15]. Follow-up research has examined language-specific contexts. Altunay et al. constructed deep learning-based English and Turkish SMS spam filter systems [6], while De Goma et al. combined TF–IDF with neural models to detect lexical and contextual features [7]. At the same time, OCR-based deep learning pipelines were proposed for detecting scams, namely for regions where scam messages resemble official messages [8]. Even such massive surveys continue to emphasize the importance of benchmark tasks and carefully filtered SMS datasets in increasing model robustness [2], [9]. In the Bangla NLP realm, particular work has emerged to address spam and smishing classification. Uddin et al. explored transformer-based approaches for Bangla SMS classification, showing that pretrained approaches are better than baselines but often necessitate large datasets and computational power [10]. Aliza et al. explored the impact of BERT and ELMo embeddings on Bangla spam classification, stressing the significance of contextual representation [11]. Parallel efforts include Tanbhir et al.'s hybrid ML models for Bangla smishing detection [12], and Johari et al.'s BangalaBarta dataset, the first open large-scale resource for Bangla spam and smishing classification [13], [20]. Cross-architecture solutions have brought forth advanced hybrid models such as ALBERT–BiLSTM with cross-attention for SMS detection [14] and ResNet- or CNN–RNN-based architectures that particularly capture local features along with sequential dependencies [15]. Johari et al. subsequently presented SpaLLM-Guard which employs open-source as well as commercial LLMs to identify SMS spam at scale [16]. These papers indicate the recent shift toward transformer and LLM architectures but also emphasize that recurrent models such as BiLSTM remain contender models for relatively smaller and well-controlled datasets such as SMS corpora. Finally, practical reasons behind Bangla SMS classification are supported by regulatory and practical concerns. The Telecom Regulatory Authority of India (TRAI), for example, mandates the classification of SMS into promotional, spam, or transactional types via suffixbased identifiers [21]. This reinforces the significance of automated classification of SMS spam/smishing as a research issue as well as a socially important technology for digital communication protection.

3 Methodology

The proposed methodology for SMS phishing detection is illustrated in Fig. 1. The workflow consists of six primary stages: (1) data collection, (2) data preprocessing, (3) model selection,(4) model evaluation, and (5) Output(Fig. 1).

3.1 Data Collection

The dataset utilized in this research was downloaded from Mendeley Data [13], which guarantees accessibility and reproducibility for subsequent work. It consists of 2,772 Bangla SMS messages that have been labeled into three distinct classes: normal, promotional, and smishing (smaphising). All the three classes contain 924 samples, so the data is balanced and avoids class imbalance duringsupervised training. The messages were collected from different reallife sources like authentic user messages, business advertisements, and inadvertently reported smishing attacks in cellular networks. Each entry in the dataset contains the raw Bangla SMS message and its class label. This dataset is a suitable foundation for Bangla SMS classification research where the task is distinguishing between real messages, advertisement content, and fraudulent smishing attempts. Its balanced nature and availability via Mendeley Data make it highly suitable for benchmarking deep models such as BiLSTM for sequence classification tasks.

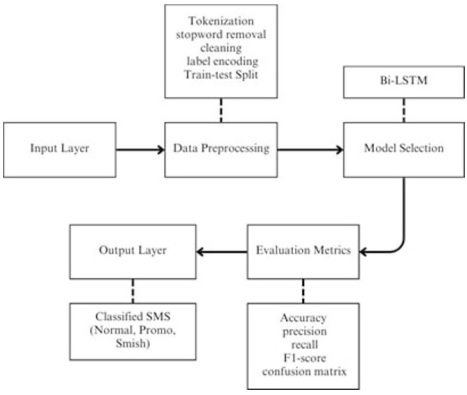

Fig. 1. Proposed Methodology of the work

Methodology adopted in this study involves several systematic steps, including dataset selection, preprocessing, tokenization, feature representation, model designing, and evaluation. Each step is executed systematically to build robust classification of Bangla SMS into Smish, Promotional, and Normal categories (Fig. 2).

3.2 Data Preprocessing

Prior to model training, a preprocessing was undertaken to maintain the data consistent and compatible with deep models. The original Bangla SMS messages were first tokenized using a word-level Keras tokenizer with a fixed 10,000-word vocabulary size. Out-of-vocabulary words were replaced with a special ¡OOV¿ token to support unseen words. For uniform input length, all sequences tokenized

were padded or trimmed to a maximum of 10 tokens to ensure efficient batch processing during training. An 80-20 split was used to create training and validation Figure 2. Sample Dataset sets comprising 2,217 messages for training and 555 messages for validation. Stratified sampling was used to preserve the class distribution between the two sets. As the data was already balanced among the three classes (normal, promotion, and smishing), no oversampling or undersampling was necessary. Class weights were, however, computed and added to the loss function in order to reduce any additional bias while optimizing. These preprocessing operations made sure that input data was balanced, normalized, and in a state to be trained for the BiLSTM classification model. /subsectionModel Selection We employed a Bidirectional Long Short-Term Memory (BiLSTM) network as the underlying classification model for our work. The decision was motivated by the suitability of sequential models in acquiring context dependence between short Bangla SMS texts. Unlike a unidirectional LSTM, which acquires knowledge from tokens in the left-to-right direction only, BiLSTM incorporates both forward and backward sequence processing, thereby allowing the model to gain knowledge from future and past context simultaneously. The architecture consisted of an embedding layer of vocabulary size 10,000 and embedding dimension 128, followed by a two-layer BiLSTM with 128 hidden units. A dropout of 0.5 was applied to prevent overfitting. The output from the BiLSTM layers was passed through a fully connected linear layer to classify into one of the three target classes: normal, promotional, or smishing. The Adam optimizer was used to train the model, with a learning rate of 0.0001. Class weights from the training dataset were added to the cross-entropy loss function to correct for bias. Training was conducted for 10 epochs with early stopping on validation loss not improving, in order to achieve generalization and avoid overfitting. By focusing on a BiLSTM classifier, this study demonstrates the ability of sequence-based deep learning models to capture fine-grained distinctions between different classes of Bangla SMS.

3.3 Model Training

The proposed BiLSTM model was trained using the Adam optimizer with a learning rate of 0.0001. The minibatch size of 32 was used for training, and training was performed to a maximum of 10 epochs. For preventing overfitting and increasing the generalization, 0.5 dropout probability was employed within recurrent layers. Since the dataset was class-balanced for normal, promotional, and smishing classes, a weighted loss function was employed to ensure that all classes were learned equally well by the model. Additionally, early stopping on validation loss was employed to terminate training as soon as the performance ceased to improve, hence reducing unnecessary computation. All experiments were conducted in a GPU-boosted configuration, which significantly accelerated the sequential

3.4 Model Evaluation

The evaluation process was such that a full picture of how the model performed was presented. In this regard, standard classification measures were calculated in the form of accuracy, precision, recall, and F1-score, as well as their complementary roles of overall correctness and class-specific performance. Confusion matrices were also created to look at the misclassifications incurred between the three SMS classes: normal, promotional, and smishing. This testing environment therefore enabled us to see in close detail how the model performed on various categories of Bangla SMS messages and pinpoint where the most significant errors, like labeling smishing as normal, occurred.

Accuracy: The overall proportion of instances that are correctly classified out of the given predictions.

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

Precision: How many of the predicted positive instances are in fact true positive instances.

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

Recall: How capable the model is of capturing all relevant(actual positive) instances.

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

F1-Score: The harmonic mean of precision and recall; it balances false positives and false negatives.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Here, TP denotes True Positives, TN denotes True Negatives, FP represents False Positives, and FN represents False Negatives.

In addition to these evaluation metrics, the Receiver Operating Characteristic (ROC) curve and the corresponding Area Under the Curve (AUC) score were also computed. The ROC–AUC serves as a comprehensive performance indicator across varying classification thresholds and is particularly valuable for visualizing the trade-off between the true positive rate (TPR) and false positive rate (FPR). For the multi-class classification task, ROC–AUC was calculated using the One-vs-Rest strategy, and both macro-averaged and micro-averaged AUC scores were reported. Furthermore, a confusion matrix was generated for the best-performing model to clearly depict the distribution of correct and incorrect predictions across the three SMS categories: normal, promotional, and smishing. This visualization provided deeper insight into misclassification patterns, especially in cases where smishing messages were incorrectly labeled as normal, highlighting the challenge of distinguishing fraudulent content from legitimate communication.

4 Results and Discussion

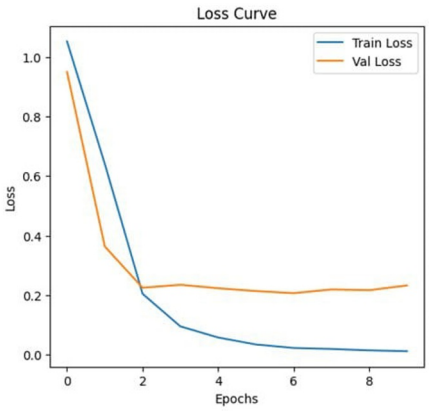

Fig. 2. Training and Validation Loss Curve

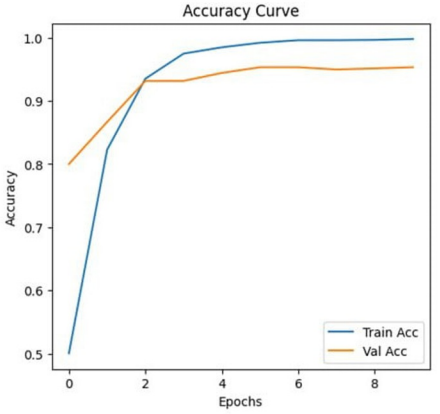

Fig. 3. Training and Validation Accuracy Curve

The loss curve (Fig. 3) shows a uniform decrease in training and validation loss over the epochs. The training loss rapidly falls in the first three epochs and keeps on converging towards near-zero values, while the validation loss tapers off at 0.2 after a rapid drop initially. This indicates that the BiLSTM model learned the patterns in the Bangla SMS data quite well with minimal overfitting. The accuracy curve (Fig. 4) also shows this result. The training accuracy rises sharply from around 50the start to nearly 100accuracy also rises steadily, to about 96stable throughout later epochs. The minor but consistent gap between training and validation accuracy shows that the model generalizes well to new SMS messages as well as possessing strong learning capacity. Simultaneously, the curves demonstrate that the BiLSTM model achieved fast convergence, steady performance, and robust generalization, appropriate for Bangla SMS classification into normal, promotion, and smishing classes. The figure (Fig. 5) presents a confusion matrix for the performance of a three-class classification model that is supposed to distinguish between normal, promo, and smish classes. The true (actual) labels are presented in the rows of the matrix, while the predicted labels by the model are presented in the columns. Accurate classifications are presented in the diagonal entries, while misclassifications are presented in the off-diagonal entries.For the normal class, the model correctly classified 181 samples, with 2 misclassified as promo and 2 as smish.

For the promo class, the model correctly identified 184 samples, with 1 misclassified as smish and none as normal. For the smish class, the model correctly classified 164 samples, but 17 were incorrectly predicted as normal and 4 as promo.The confusion matrix also includes a color-coded heatmap, where darker

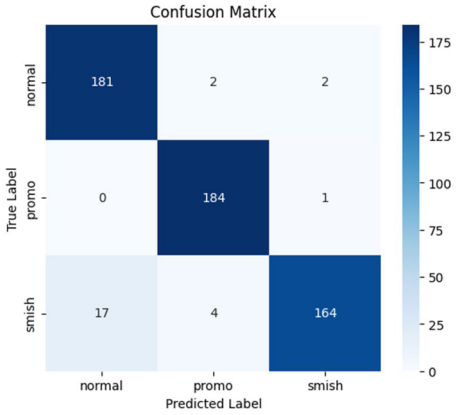

Fig. 4. Confusion Matrix of Bi-LSTM

blues indicate higher classification frequencies. A color bar to the right provides a numerical scale for shade intensity interpretation, ranging from light (low frequency) to dark (high frequency). This matrix indicates the classifier performing well across all three classes, with very high accuracy for both promo and normal messages. There is, however, a high level of confusion between the smish and normal classes, as seen by the 17 misclassifications, which shows that the model has the most difficulty distinguishing between these classes.

Table 1. Bi-LSTM Model Classification Report

Class	Precision	Recall	F1-Score	Support
Normal	0.91	0.98	0.95	185
Promo	0.97	0.99	0.98	185
Smish	0.98	0.89	0.93	185
Accuracy			0.95	555
Macro Avg	0.95	0.95	0.95	555
Weighted Avg	0.95	0.95	0.95	555

Classification report (Table. I) shows the performance measurement of a three-class classifier to distinguish between normal, promo, and smish message classes. Metrics reported are precision, recall, F1-score, and support, computed for each individual class, along with overall accuracy, macro average, and weighted average results.

For the normal class, the classifier precision was 0.91, i.e., 91% of the instances predicted as normal were correctly classified. Recall of 0.98 shows that the model correctly identified 98% of all true normal samples. Consequently, the F1-score, which is a balance between precision and recall, was 0.95.

For the promo class, the model had precision of 0.97 and recall of 0.99, for an F1-score of 0.98. This means that the classifier is extremely good at both finding and correctly classifying promotional messages, with incorrect classifications being very rare.

For the smish class, the classifier achieved a precision of 0.98, showing that nearly all the predictions made as smish were accurate. The recall dropped to 0.89, showing that 11% of smish messages were misclassified into the other classes. The F1-score of 0.93 shows a slight performance drop for this class compared to the others.

At the system level, the classifier was 95% accurate in that 95% of the total 555 instances in all classes were correctly classified. The macro-average precision, recall, and F1-score of 0.95 each indicates the model has a balanced performance across all three classes without bias towards any particular category. Similarly, the weighted-average values of 0.95 also indicate that class distribution did not play a significant role in performance, pointing towards robustness in multi-class classification.

This report highlights that the classifier works very well overall, with particularly good results for the promo and normal classes. The primary limitation is the slightly lower recall for the smish class, which is consistent with the findings from the confusion matrix where some smish samples were misclassified as normal.

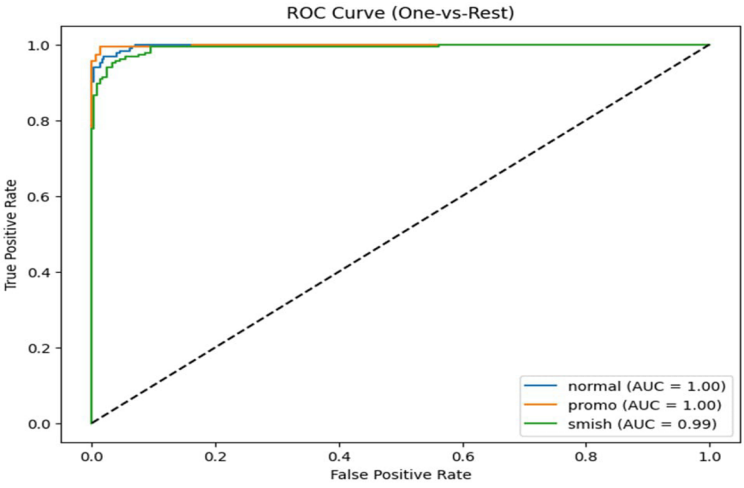

Fig. 5. ROC–AUC curve for Bi-LSTM model

The graph Fig. 5 displays the Receiver Operating Characteristic (ROC) curves for a multi-class classification problem with One-vs-Rest (OvR) strategy. The one for each is the trade-off between the True Positive Rate (TPR), or sensitivity, on the y-axis and the False Positive Rate (FPR) on the x-axis for a single class. The diagonal dashed line is the baseline for random classification with Area Under the Curve (AUC) of 0.5.

The ROC plots indicate perfect discrimination ability for all three classes. The curve for the normal class (blue) touches the top-left corner with AUC = 1.00, which is the classification with zero overlap between positive and negative classes. Similarly, the promo class (orange) also has AUC = 1.00, confirming that the classifier perfectly discriminates promotional messages from the other classes.

The smish class (plotted in green) has an AUC of 0.99 with highly excellent separability, though marginally lower than the last two classes. The ROC curve of smish is very close to the top-left boundary with hardly any deviation.

Generally, the ROC analysis demonstrates that the classifier is remarkably high in performance for all classes. The ideal AUC values for normal and promo indicate that the model classifies no incorrectly for these classes across different decision thresholds. The lower AUC for smish confirms earlier findings from the confusion matrix and classification report, where the classifier showed smaller misclassifications between smish and normal.

5 Conclusion and Future Work

This work introduced a BiLSTM-based method to detect smishing in Bangla, using a class-balanced dataset of 2,772 SMS messages from normal, promotional, and smishing classes. Preprocessing, tokenization, and padding served as inputs of the same format, while classweighted loss and early stopping facilitated stable training.

The model was 95% accurate overall, with high precision and recall for promotional and normal SMS. Detection of smishing was harder, with recall declining to 0.89 because of misclassification as normal messages. ROC analysis supported the model's resilience, with AUCs of 1.00 for normal and promotional, and 0.99 for smishing.

These results reflect the effectiveness of BiLSTM models in Bangla SMS classification and highlight the imperative for stronger methods (e.g., transformer, attention, or focal loss) to further reduce false negatives in smishing. This work offers both a benchmark for Bangla NLP research and an applied foundation for mobile communication security, with subsequent research focused on better architectures and real-time application.

References

1. A. Aliza et al.: A Comparative Analysis of SMS Spam Detection Employing Machine Learning Methods. In: Proc. 6th Int. Conf. on Computing Methodologies and Communication (2022).
2. M. R. Al Saidat et al.: A Comprehensive Survey of NLP and ML Techniques for SMS Spam Detection. Procedia Computer Science, vol. 187 (2024).
3. G. Airlangga: A Comparative Study of MLP, CNN, LSTM, BiLSTM, GRU, and Attention Mechanisms for SMS Spam Detection. Malang Computer Science and Engineering Journal, vol. 6, no. 4 (2024).
4. S. Kaddoura et al.: A Systematic Literature Review on Spam Content Detection in Social Media. Frontiers in Computer Science, vol. 4, no. 1 (2022).
5. O. Abayomi-Alli et al.: A Deep Learning Method for Automatic SMS Spam Detection. Concurrency and Computation: Practice and Experience, vol. 34, no. 2 (2022).
6. H. C. Altunay et al.: SMS Spam Detection System Based on Deep Learning Architectures for Turkish and English Messages. Applied Sciences, vol. 14, no. 24 (2024).
7. J. De Goma et al.: Detection of SMS Spam Messages Using TF-IDF Vectorizer and Deep Learning. In: Proc. 2024 ACM Conf. on Artificial Intelligence and Data Science (2024).
8. A. Shinde et al.: SMS Scam Detection Application Based on Optical Character Recognition and Deep Learning. Journal of Medical Systems, vol. 48, no. 5 (2024).
9. M. S. Johari et al.: Key Insights into Recommended SMS Spam Detection Datasets. Scientific Reports, vol. 15, no. 1 (2025).
10. M. A. Uddin et al.: Exploring Transformer-Based Language Modeling Approaches for Bangla SMS Spam Detection. Procedia Computer Science, vol. 187 (2025).
11. M. R. Aliza et al.: Exploring BERT and ELMo for Bangla Spam SMS Dataset Creation and Detection. In: Proc. 2024 Int. Conf. on Natural Language Processing (2024).
12. G. Tanbhir et al.: Hybrid Machine Learning Model for Detecting Bangla Smishing SMS. arXiv preprint arXiv:2502.01518 (2025).
13. M. S. Johari et al.: BangalaBarta: A Spam/Smishing SMS Dataset for Bangla. Mendeley Data (2025).
14. B. S. Aparna et al.: ALBERT-BiLSTM Cross-Attention Network with Progressive Learning for SMS Spam Detection. Journal of King Saud University - Computer and Information Sciences (2025).
15. G. Airlangga: A Comparative Analysis of Deep Learning Models for SMS Spam Detection: CNN-LSTM, CNN-GRU, and ResNet Approaches. CNAPC Journal, vol. 6, no. 4 (2024).
16. M. S. Johari et al.: SpaLLM-Guard: Pairing SMS Spam Detection Using Open-source and Commercial Large Language Models. arXiv preprint arXiv:2501.04985 (2025).
17. M. R. Aliza et al.: A Comparative Analysis of SMS Spam Detection Employing Machine Learning Methods. In: Proc. 6th Int. Conf. on Computing Methodologies and Communication (2022).
18. M. S. Johari et al.: Key Insights into Recommended SMS Spam Detection Datasets. Scientific Reports, vol. 15, no. 1 (2025).
19. G. Tanbhir et al.: Hybrid Machine Learning Model for Detecting Bangla Smishing SMS. arXiv preprint arXiv:2502.01518 (2025).

20. M. S. Johari et al.: BangalaBarta: A Spam/Smishing SMS Dataset for Bangla. Mendeley Data (2025).
21. Telecom Regulatory Authority of India: Fed up with Pesky SMS? Now You Can Spot Spam, Promo, or Important Messages by Checking the P, S, T, or G Suffix. The Economic Times, May 6 (2025).