Explainable Hybrid Deep Learning for Advanced Bengali SMS Classification: An XLMR and LSTM Approach

Manash Sarker, Ahammad Hossen Eazaz, Md Intishar Ahmed, Golam Md. Muradul Bashir

Department of Computer and Communication Engineering

Correspondence: intishar17@cse.pstu.ac.bd

(Received Date: 10th March 2026, Accepted Date: 1st July 2026)

Abstract: The recent development of Artificial Intelligence (AI) has significantly impacted the classification of SMS messages and the detection of smishing attacks, which have become prominent concerns given the rise of fraudulent messaging. These attacks pose a serious threat to individual security, as fraudulent SMS messages can lead to identity theft. Traditional machine learning models have struggled to cope with the complexity of SMS text, often failing to produce satisfactory results. This study addresses these limitations by developing a Hybrid XLM-R-LSTM model, in which contextualized transformer embeddings from XLM-RoBERTa are combined with an LSTM layer to analyze Bengali text. Under a stratified internal train-test evaluation, the model achieved an accuracy of 99.48% and a weighted F1-score of 0.9948 on a dataset of 8,629 SMS messages; external and cross-domain validation remain directions for future work. The proposed model outperformed traditional machine learning baselines (Random Forest: 98.19%, XGBoost: 97.48%, Logistic Regression: 97.29%) as well as contemporary deep learning models (mBERT, BanglaBERT, XLM-RoBERTa). SHAP (SHapley Additive exPlanations) was used to interpret the model's output, illustrating how individual words contribute to each prediction. The model is efficient enough for deployment in low-resource environments, requiring only 12 GB of RAM and 15 GB of GPU memory.

Keywords: Smishing Detection, Bengali Language, Hybrid Model, Deep Learning, Explainable AI, XLM-RoBERTa, LSTM.

Introduction

The contemporary cyber landscape has undergone a paradigm shift with the widespread adoption of Artificial Intelligence (AI) technologies. While this has strengthened cyber-defense mechanisms, it has simultaneously equipped cybercriminals with the ability to carry out increasingly sophisticated social engineering attacks (Sarker, 2021). Among the threats to emerge from this shift, “smishing” — the use of SMS messages for phishing — has become one of the most prominent means of conducting large-scale data theft and financial fraud (Alsaleh et al., 2021). Recent industry reporting for 2025 indicates a sharp rise in AI-driven phishing attempts, a trend attributed to the growing availability of Large Language Models (LLMs) that allow attackers to generate convincing fraudulent content at very low cost (Federal Trade Commission, 2024; Sasa Software, 2025). The scale of the problem is considerable: smishing alone is estimated to account for roughly one-third of all phishing attempts worldwide, contributing to billions of dollars in annual financial losses (Federal Trade Commission, 2024; APWG, 2025).

In Bangladesh specifically, this risk is amplified by the fact that the number of active mobile subscriptions exceeds the national population, with tens of millions of users routinely relying on SMS for financial and government-related transactions (BTRC, 2024). Attackers frequently bypass conventional security filters by exploiting the structural complexity of the Bengali language, including “Banglish” transliteration and code-mixed dialects that rule-based filters struggle to detect (Joshi et al., 2020; Tanbhir et al., 2022). Progress toward more robust defenses is further constrained by Bengali’s status as a low-resource language, where the scarcity of high-quality labeled data limits the development of effective detection systems (Joshi et al., 2020; Shahriyar and Tanbhir, 2025).

Detection methods for SMS security have evolved from rigid keyword-based rules toward more adaptive machine learning (ML) approaches (Gedam & Banchhor, 2024; Sarker et al., 2020). Early models built on Support Vector Machines (SVM) and Naïve Bayes offered a useful baseline but depended heavily on manual feature engineering (Ahmed et al., 2021; Robert et al., 2023). The subsequent shift toward neural architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) improved the modeling of sequential dependencies within deceptive narratives (Khan et al., 2019; Ghourabi et al., 2020; Al-

Qurishi et al., 2021). Even so, these architectures remained limited in capturing long-range dependencies until the introduction of the Transformer architecture (Vaswani et al., 2017).

Transformer-based architectures, including BERT and RoBERTa, now dominate NLP research, employing self-attention mechanisms to model text bidirectionally (Devlin et al., 2018; Liu et al., 2019). Multilingual extensions such as XLM-RoBERTa build on advances in cross-lingual transfer learning (Artetxe et al., 2020) and have demonstrated strong generalization across diverse linguistic settings (Conneau et al., 2019). For Bengali specifically, recent work has focused on fine-tuning these models or combining them into hybrid pipelines to address the language's morphological complexity (Saha & Nanda, 2023; Tanbhir et al., 2025; Uddin et al., 2024). However, purely contextual models can still under-represent fine-grained temporal cues present in deceptive communication (Tan et al., 2022; Ragab et al., 2025), suggesting that combining the global semantic strength of Transformers with the sequential focus of LSTM networks may offer a meaningful advantage for threat detection (Zhao et al., 2025; Kabir et al., 2025).

Beyond predictive accuracy, the "black-box" nature of modern neural architectures remains a significant barrier to real-world deployment (Olasehinde, 2024; Simon, 2025), motivating growing interest in Explainable AI (XAI) methods that justify model decisions (Guidotti et al., 2018). SHAP and LIME are widely used for this purpose, providing word-level attribution for model predictions (Lundberg and Lee, 2017; Ribeiro et al., 2016). While Transformer-based sentiment analysis has been explored for Bengali, the combination of a Hybrid XLM-R-LSTM architecture with XAI in the specific context of Bengali SMS classification remains largely unexplored (Chinnasamy et al., 2024; Jain et al., 2025; Hasan et al., 2025).

This research addresses this gap by proposing an interpretable deep learning pipeline. The main contributions of the proposed work are a new framework for multiclass categorization, comprising Normal, Spam, and Promotional categories, based on specific linguistic patterns of the Bengali language. The proposal of a Hybrid XLM-R-LSTM model that jointly leverages the contextual and sequential understanding of the input text. The evaluation of the proposed model against multiple baselines, including the commonly used BanglaBERT and mBERT models. The application of a SHAP-based visual attribution approach to ensure the interpretability of the proposed model for security practitioners.

Methodology

Data Collection

The dataset of Bengali SMS messages used for classification was carefully constructed to ensure adequate diversity and representation. The base dataset was compiled from a pool of 2,200 messages obtained from IEEE Dataport (Tanbhir et al., 2024). For additional diversity, further SMS messages were collected across categories such as spam, promotional, and smishing content, including messages retrieved from online banking platforms. This ensured that the dataset represented a wide spectrum of SMS messages likely to be encountered in real-world scenarios.

Data augmentation techniques, such as replacing Bengali numerals with English numerals and introducing textual noise, were applied to increase the size and variety of the training data. The resulting dataset consists of over 8,000 SMS messages spanning a diverse range of categories, as shown in Fig. 1, which improves the robustness of the classification model.

Label	Textual Description
Normal	সাফল্য তোমার দিকে আসবে, ধৈর্য ধরো।
Promo	অনলাইন শপিংয়ে ২০% পর্যন্ত ডিসকাউন্ট! কোড ব্যবহার করে সুবিধা নিন।
Smish	আপনার অ্যাকাউন্টে সন্দেহজনক কার্যকলাপ হয়েছে। লগইন করে যাচাই করুন: [accountverify.com]

Figure 1. Sample data of SMS message

Of the final 8,629 messages, approximately 4,500 were sourced directly from IEEE Dataport and additional manual collection, while approximately 3,500 were generated through augmentation of the original messages. These SMS messages are classified into three types: smish, normal, and promo. Fig. 2 shows the distribution of messages across these three types, with the highest number recorded in the smish category (3,313 messages), followed by normal (2,666 messages) and promo (2,650 messages). Each type is assigned a distinct color: blue for smish, green for normal, and red for promo.

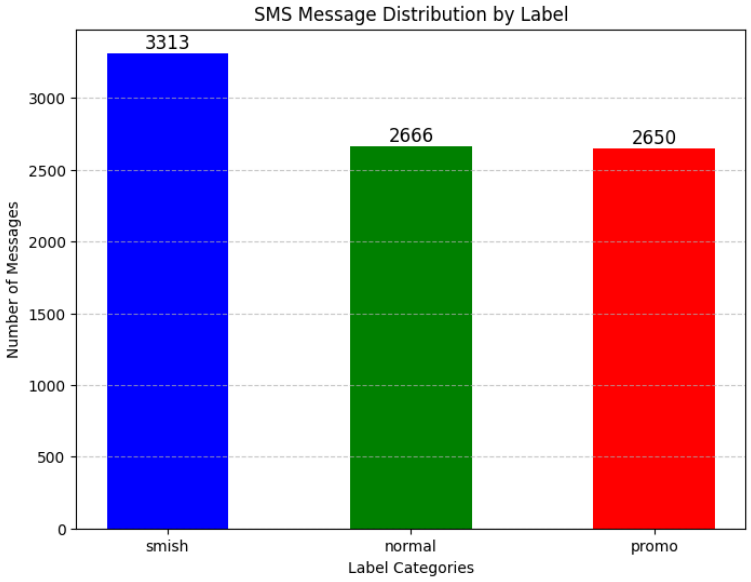

Figure 2. SMS message distribution in the dataset

Data Quality and Leakage Consideration

Repeated patterns, forwarded promotional text, and highly similar messages are a known risk factor for data leakage in SMS classification tasks. To mitigate this risk, the dataset was reviewed to remove obvious repetitions, and stratified sampling was used to keep class proportions (smish, promo, normal) consistent across the training and test sets.

It should be noted that the train-test split was stratified by class label only, not by message origin or augmentation group. A full discussion of the resulting leakage risk, its severity, and how it is addressed is provided in the Limitations section.

Data Preprocessing & Splitting

An important preprocessing step involved leveraging the tokenizer of the pre-trained XLM-RoBERTa model. This tokenizer employs a subword segmentation method known as SentencePiece, which performed well on this dataset. Rather than relying on a fixed dictionary, the SentencePiece tokenizer intelligently segments the input text, including slang, common misspellings, and the hybrid Banglish script. Each message was converted into a sequence of numerical tokens, with special tokens marking the start and end of the message. Messages were then padded to a standard length to enable batch processing, and an attention mask was generated for each message so the model would focus only on genuine content while ignoring padded tokens. This tokenizer played a vital role in enabling the model to learn the intricacies of the Bengali language from the raw SMS dataset. To encode the message types, the classes Smish, Promo, and Normal were represented by the numbers 0, 1, and 2, respectively.

The preprocessed dataset of 8,629 messages was split 80-20 for training and testing while keeping class proportions balanced. Specifically, the Smish class comprised 2,650 training and 663 testing messages, the Promo class comprised 2,120 training and 530 testing messages, and the Normal class comprised 2,133 training and 533 testing messages, as shown in Table 1.

Table 1. Class-wise distribution of training and testing samples

Class	Training Samples	Testing Samples
Smishing (smish)	2,650	663
Promotional (promo)	2,120	530
Normal (normal)	2,133	533
Total	6,903	1,726

Proposed Custom Model: Hybrid-XLM-R-LSTM

Our contribution is a hybrid architecture that augments a pre-trained XLM-RoBERTa Transformer model with an LSTM layer to jointly process contextual and sequential information. The architecture is shown in Fig. 3 and consists of the following blocks:

Contextual Embedding Layer (XLM-RoBERTa): The raw text is first tokenized and passed into XLM-RoBERTa, a multilingual Transformer-based text encoder. Using self-attention, the model computes attention scores between word pairs, allowing it to capture context-dependent relationships across a message. It maps the text into a dense vector representation while preserving semantic meaning, enabling the model to capture the subtle nuances that determine how a given keyword should be interpreted within a specific message.

Sequential Pattern Learning Layer (LSTM): The resulting embeddings are passed into an LSTM layer (Hochreiter and Schmidhuber, 1997), which is well suited to capturing long-range dependencies between words in a sequence. This allows the model to track the order in which words appear within a message, helping it recognize time-sensitive or sequential cues that are often present in smishing and promotional content.

Classification Layer (Dense + Softmax): This block consists of a dense neural layer followed by a softmax function, which determines the probability of a message belonging to each class (“Smish,” “Promo,” or “Normal”). The predicted class is the one with the highest probability score.

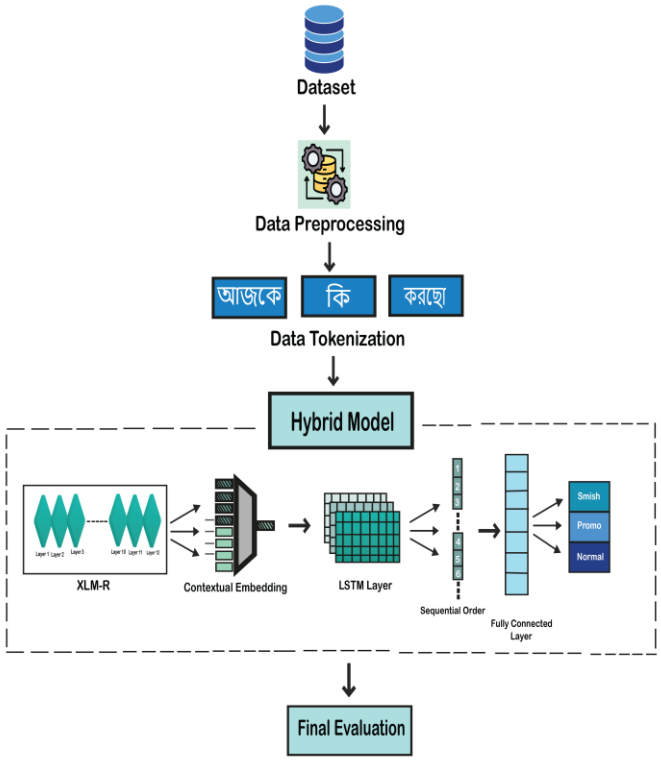

Figure 3. Architecture of the proposed Hybrid-XLM-R-LSTM model. (Authors' own design)

The proposed Hybrid-XLM-R-LSTM overall model architecture is illustrated which starts with the input of a dataset and its preprocessing, followed by tokenization of Bangla text, embedding in the context using XLM-RoBERTa, sequential pattern learning using LSTM and finally classification into three classes Smish, Promo and Normal using a dense-softmax layer.

Experimental Design & Training Process

To achieve an unbiased evaluation of the model's performance, the dataset was divided into a training set (80%, 6,903 messages) and a held-out test set (20%, 1,726 messages). A 2-fold stratified cross-validation method was used to identify the best-performing architecture. All reported metrics reflect performance within the same data distribution and collection period as the training set; the implications of this internal-validation-only setup are discussed further in the Limitations section.

Models were implemented using the PyTorch deep learning framework and trained in the Google Colab environment using a T4 GPU with 16 GB of VRAM. The AdamW optimizer was used with a batch size of 32, for a maximum of 10 epochs. An early-stopping callback halted training if the F1-score did not improve for two consecutive epochs, in order to prevent overfitting, and save_total_limit=1 was used to retain only the best-performing checkpoint given limited storage space.

Workflow of Hybrid-XLM-R-LSTM Model

The workflow of the model, depicted in Figure 4, is structured as follows:

Figure 4. Workflow of the proposed Hybrid-XLM-R-LSTM model. **(**Authors' own design)

As shown in this flowchart, an SMS text classification system was created by combining the feature extraction pipeline of XLM-RoBERTa modality with an LSTM classifier and an interpretability pipeline of SHAP for the classification of SMS. The input SMS text is subjected to data preprocessing and then it takes two parallel workflows. For data handling, the labels are subjected to label encoding, after which they are split into balanced and robust trains and tests sets, using stratified train-test split and K-Fold Cross-Validation. Meanwhile, model building branch, the text is tokenized into words and then features are extracted with XLM-RoBERTa feature extraction, passed through an LSTM Layer for sequence learning and finally through a fully connected output layer to get classification logits. The two branches merge at the training process stage, where the model is trained with the data prepared and the features extracted. The trained model is then passed to model evaluation, which in turn branches to two paths: 1) SHAP analysis model interpretation and 2) SHAP plot generation, which is for interpretation of feature contributions and model decisions, or directly to prediction output, which is for the model's classification results. Finally, the Interpretability results are merged into a single report & model insights phase, which then passes to the final phase of model optimization for continuous improvement of the model.

Results and Discussion

Comparative Performance

As shown in Table 2, several conventional machine learning models — Random Forest, XGBoost, Logistic Regression, Linear SVM, Multinomial Naive Bayes, and K-Nearest Neighbors — were evaluated as baselines. While these models performed reasonably well, none outperformed the proposed Hybrid-XLM-R-LSTM model on the SMS classification task. To the best of our knowledge, this is the first study to benchmark this dataset against deep learning models such as mBERT, XLM-RoBERTa, and BanglaBERT.

Table 2. Performance comparison of different models

Model	Accuracy (%)
Random Forest	98.19
XGBoost	97.48
Logistic Regression	97.29
Linear SVM	97.12
Multinomial Naive Bayes	94.41
K-Nearest Neighbors	76.04
mBERT	99.44
XLM-RoBERTa (Base)	99.43
BanglaBERT	99.39
Hybrid-XLM-R-LSTM (Ours)	99.48

The cross-validation results in Table 3 show that Hybrid-XLM-R-LSTM performed best, with an average F1-score of 0.99478 and a standard deviation of 0.0014.

Table 3. Cross-validation performance comparison

Model	Avg F1 (CV)	Std F1 (CV)
BanglaBERT	0.993915	0.000871
XLM-RoBERTa	0.994353	0.000871
mBERT	0.994496	0.000144
Hybrid XLM-R + LSTM	0.994786	0.001448

Evaluation on Held-Out Test Set

The final model was evaluated on the unseen test set of 1,726 messages, achieving an accuracy of 99.48% and a macro-averaged F1-score of 0.9948, as detailed in Table 4.

Table 4. Classification report on the test set

Class	Precision	Recall	F1-Score	Support
Smish	0.99	1.00	1.00	663
Promo	1.00	0.99	1.00	530
Normal	1.00	1.00	1.00	533
Accuracy			0.9948	1,726
Macro avg	0.99	1.00	1.00	1,726

Although the reported metrics round to 1.00 at one decimal place for most classes, the model did not achieve perfect classification: six of the 1,726 test messages were misclassified, as shown in the confusion matrix (Figure 5).

Robustness Analysis of the High Classification Accuracy

Given the notably high classification performance achieved by the proposed Hybrid-XLM-R-LSTM model, it is important to critically assess whether the reported result is genuinely generalizable or is instead influenced by overfitting or accidental memorization. Several pieces of evidence support the former. First, the model did not achieve perfect classification: six of 1,726 held-out test messages were misclassified (Fig. 5), indicating that predictions are not simply memorized outputs. Second, no explicit regularization beyond early stopping was required to reach this level of performance — training was halted automatically once the F1-score failed to improve for two consecutive epochs, which limits the model's opportunity to overfit to the training partition. Third, the cross-validation results (Table 3) show an average F1-score of 0.994786 with a low standard deviation of 0.001448 across folds, indicating that performance is stable across different partitions of the data rather than being an artifact of a single train-test split.

It is also worth noting that several baseline models without contextual embeddings (e.g., Logistic Regression at 97.29%, Random Forest at 98.19%) achieved comparably high accuracy (Table 2). This pattern suggests that Bengali smishing, promotional, and normal SMS messages in this dataset are separable through strong lexical and structural cues — such as transactional URLs, verification keywords, discount codes, and templated banking phrases — rather than requiring deep contextual reasoning alone, a pattern of lexical regularity previously noted in SMS spam/phishing corpora (Sarker et al., 2020). Taken together with the leakage considerations discussed in the Limitations section, this indicates that the reported accuracy figures should be read as indicative of strong performance under the current experimental setup, rather than as a precise measure of real-world detection difficulty.

A related concern is the risk posed by high-confidence misclassification, illustrated by the example in Fig. 10, where a legitimate promotional message was misclassified as smishing with 99.59% confidence. Unlike low-confidence errors, which are relatively easy to flag for manual review, high-confidence errors are more likely to be trusted and acted upon without further scrutiny in a deployed system. This has direct practical implications: a deployment pipeline built on this model should not rely on the model's confidence score alone to decide when human review is required, since confidence and correctness can diverge sharply on messages containing banking- or service-related terminology, as this example shows. Mitigating this risk in future work could involve confidence-calibration techniques (e.g., temperature scaling), targeted augmentation with legitimate banking or promotional messages that resemble smishing lexically, or a human-in-the-loop review step for messages that are flagged with high confidence but originate from verified sources.

For a full discussion of the internal-validation scope of these results and directions for external validation, see the Limitations section.

Visual Analysis and Error Analysis

This is confirmed by the confusion matrix in Fig. 5, where the model's strong performance is visually confirmed, with only six misclassifications among 1,726 samples. A qualitative analysis of the misclassifications showed that they mainly occurred in situations with a high degree of ambiguity, such as promotional messages designed to resemble personal notifications. The stability of the model's performance over time is confirmed by Fig. 6, where precision, recall, and F1-score are plotted across seven training epochs.

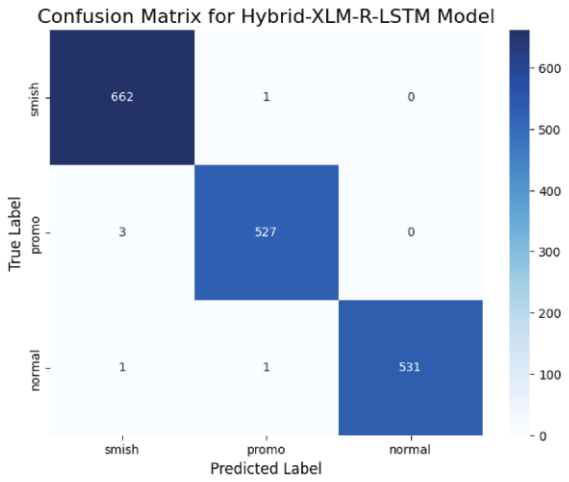

Figure 5. Confusion matrix for Hybrid-XLM-R-LSTM on the test set.

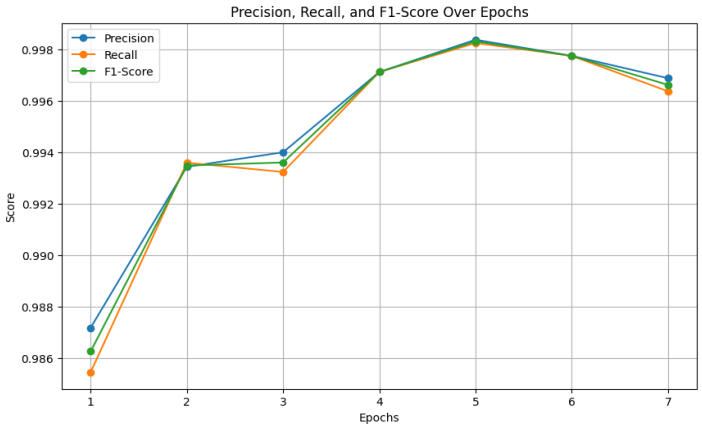

Figure 6. Precision, recall, and F1-score trends across epochs.

Exploring Model Predictions through SHAP Analysis

To move beyond the 'black-box' nature of the model and understand the reasoning behind its predictions, SHAP was used. SHAP (SHapley Additive exPlanations) is a game-theoretic approach that assigns a value to each feature — in this case, each word token — for a given prediction, providing a transparent explanation for how a particular SMS message was classified. SHAP explanations were generated for a sample of the test set for each class.

Analysis of a Smish Message: Fig. 7 shows the SHAP force plot for a correctly classified "smishing" message. The plot illustrates how individual words shift the model's prediction away from the base value of 0.00062139 toward the final output. Words such as "WhatsApp" and "স্মিশিং" (smishing) contribute strongly positive SHAP values, pushing the prediction toward the correct "smish" label.

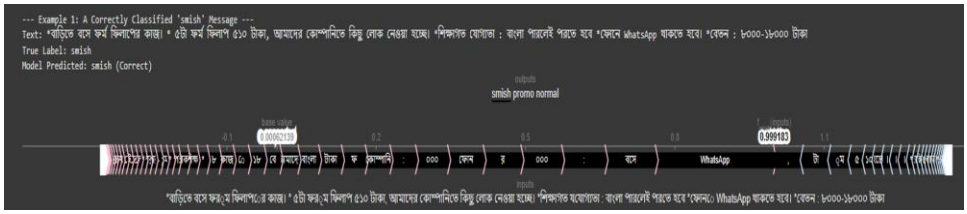

Figure 7. SHAP analysis of a smishing message

Analysis of a Promotional Message: Fig. 8 shows the SHAP analysis for a correctly classified "promotional" message. The message, containing the phrase "ফ্ল্যাট ১৫% ছাড় ডিসকাউন্ট সব শপিংয়ে।কো·ড: DISCOUNT15" (translated as "Flat 15% discount on all shopping. Code: DISCOUNT15"), was correctly labeled as "promo." The base value before feature contributions was 0.000873353. Key terms such as "DISCOUNT15" and "ফ্ল্যাট ১৫%" significantly influenced the prediction, yielding a final confidence of 0.99886. This example demonstrates the model's transparency, highlighting the importance of specific words and phrases in identifying promotional content.

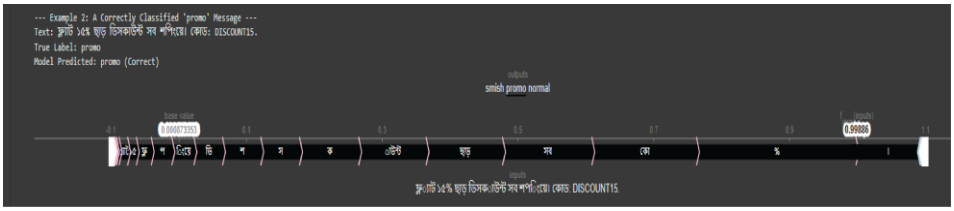

Figure 8. SHAP analysis of a promotional message

Analysis of a Normal Message: As shown in Figure 9, the model correctly classified the message as "normal." Words such as "দুঃখিত" (sorry) and "ফোন" (call) were the key contributing factors, with the model reaching a high confidence of 0.998805 relative to a baseline value of 0.000988053 before word contributions were applied.

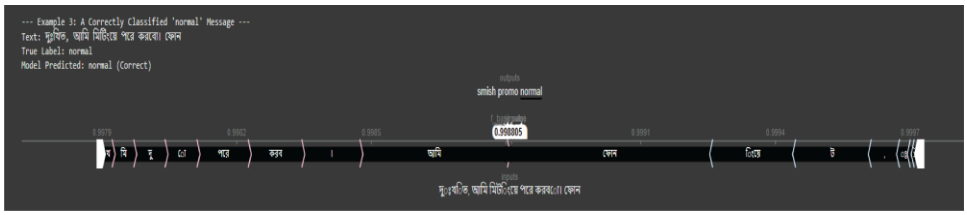

Figure 9: SHAP analysis of a normal message

Analysis of a Misclassified Message: Figure 10 presents the SHAP analysis for a case in which the Hybrid-XLM-R-LSTM model was incorrect. The input message — "হ্যাপি নিউ ইয়ার - ২০২৪ স্মার্টব্যাংকিং সেবায় সোনালী ব্যাংক সর্বদাই আপনার পাশে মোঃ আফজাল করিম সহিও এন্ড এমডি" — was a legitimate promotional message, yet the model flagged it as smishing. Notably, words such as "ব্যাংকিং" (banking), "সেবার" (service), "রেজিস্ট্রেশন" (registration), and "মোবাইল" (mobile) were all present and contributing to the prediction — terms one might expect to steer the model toward the correct label. Instead, the model misclassified the message with striking confidence, assigning a smishing probability of 0.995877. This suggests that the model has learned to associate certain banking- and service-related terms with phishing, making it difficult in some cases to distinguish legitimate promotional material from genuine phishing attempts. Practical implications of this failure mode, and possible mitigations, are discussed above under Robustness Analysis.

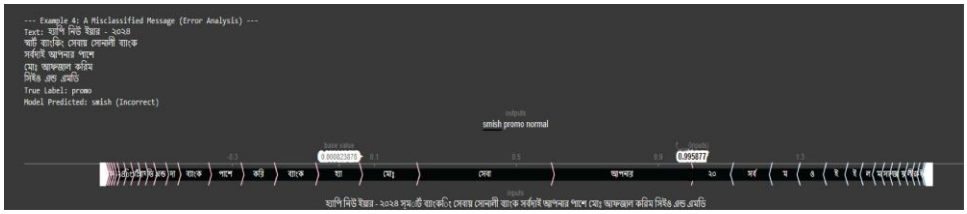

Figure 10. SHAP analysis of a misclassified message

This level of interpretability is intended not only to validate the model's learning process, but also to provide useful information to security analysts, supporting its use as an effective tool for deployment.

Limitations

Internal Validity: The train-test split in this study was stratified by class label but not by message origin or augmentation group. The augmentation process used produces linguistically distinct messages rather than exact or near-exact duplicates — variation ranges from numeral/format substitution to full sentence-level rewording — so literal text-level leakage is unlikely. However, because splitting was not origin-aware, an augmented message and its source message could still appear across both splits, and such pairs may share the same underlying intent or entities despite differing wording. This represents a milder, template-level leakage risk, distinct from literal duplication.

External Validity: The model was evaluated only on internally held-out data drawn from the same collection period and sources as the training set. No independent, temporally newer, or cross-source Bengali smishing dataset was used. Therefore, the 99.48% accuracy reported here should be understood as strong performance under internal

validation, not as evidence of generalization to real-world deployment conditions across institutions, time periods, or attack campaigns.

Construct Validity: The high accuracy achieved by both classical and transformer-based models (Table 2) suggests that the classification task, as constructed from this dataset, may be partly solvable through lexical and structural shortcuts rather than requiring full semantic understanding of deceptive intent. This raises the question of whether accuracy on this dataset fully captures the difficulty of real-world Bengali smishing detection, where attackers continuously vary their language to evade such cues.

Future work will address these threats by collecting temporally newer SMS messages from multiple independent sources, applying group-wise splitting to separate original and augmented variants, and evaluating the model on external Bengali smishing datasets and cross-domain sources (banking, promotional, personal, government) when available.

Conclusion

This paper proposes the Hybrid XLM-R-LSTM model, a deep learning approach for classifying Bengali SMS messages into three categories: smishing, promotional, and normal. Under the internal stratified evaluation protocol used in this study, the proposed model achieved a high accuracy of 99.48% on the classification task. Training was performed on Google Colab using a T4 GPU, demonstrating the feasibility of the proposed approach without the need for high-end hardware.

Beyond its overall accuracy, the model showed consistent improvement in F1-score during training, indicating that it learns the linguistic nuances that distinguish phishing from non-phishing messages. The addition of SHAP-based explainability adds further value by highlighting which words most strongly influence each prediction.

A notable aspect of this work is the environment in which it was carried out: Bengali is a low-resource language with limited annotated data and NLP tooling, yet the model performed well on a dataset of over 8,600 real-world messages. This suggests that high-performance smishing detection is achievable even in low-resource settings. Looking forward, the proposed framework has considerable potential for mobile deployment and application to other low-resource languages, offering a useful benchmark for future work in multilingual cybersecurity and explainable AI.

Conflicts of Interest: The authors declare no competing interests.

References

Ahmed, F., et al., 2021. Machine learning for SMS spam detection: A comprehensive survey. IEEE Access, 9, pp. 245-260.

Al-Qurishi, M., et al., 2021. LSTM-based deep learning for phishing detection. Security and Communication Networks, 2021, p. 5567.

Alsaleh, A., et al., 2021. A survey on smishing attacks. Journal of Information Security and Applications, 58, p. 102763.

Anti-Phishing Working Group (APWG), 2025. Phishing Activity Trends Report – 1Q 2025. Published Online.

Artetxe, M., et al., 2020. Cross-lingual transfer learning for NLP. Proceedings of ACL, pp. 120-135.

Bangladesh Telecommunication Regulatory Commission (BTRC), 2024. Mobile Phone Subscribers in Bangladesh. Available: https://www.btrc.gov.bd.

Chinnasamy, P., et al., 2024. Advanced systems leveraging AI for phishing prevention. AJSTD, 41(1), pp. 88-102.

Conneau, A., et al., 2019. Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116.

Devlin, J., et al., 2018. BERT: Pre-training of deep bidirectional transformers. arXiv:1810.04805.

Federal Trade Commission (FTC), 2024. Nationwide Fraud Losses Report 2023. Published Online.

Gedam, R. H., and Banchhor, S. K., 2024. SMS Spam Detection Using Machine Learning. JoCAAA, 33(4), pp. 434–444.

Ghourabi, A., et al., 2020. A hybrid CNN-LSTM model for SMS spam detection. Future Internet, 12(9), p. 156.

Guidotti, R., et al., 2018. A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), pp. 1-42.

Hasan, M., et al., 2025. Benchmarking Bangla Transformers for Sentiment Analysis. Int. Conf. on Language Tech., pp. 12-25.

Hochreiter, S., and Schmidhuber, J., 1997. Long short-term memory. Neural Computation, 9(8), pp. 1735–1780.

Jain, A. K., et al., 2025. NirapodBarta: A Hybrid BERT-XGBoost Model for Multiclass Bangla Smishing Detection. ResearchGate Preprint.

Joshi, P., et al., 2020. The state and fate of linguistic diversity in the NLP world. Proc. of ACL, pp. 6282–6293.

Kabir, M. R., et al., 2025. LSTM-Transformer-Based Robust Hybrid Deep Learning Model. MDPI Applied System Innovation, 7(1), p. 15.

Khan, M. A., et al., 2019. Smishing detection using deep learning with LSTM and GRU. IEEE ICICT, pp. 1–6.

Liu, Y., et al., 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.

Lundberg, S. M., and Lee, S. I., 2017. A unified approach to interpreting model predictions. NIPS 2017, pp. 4765–4774.

Olasehinde, T., 2024. Explainable AI in Phishing and Social Engineering Detection. ResearchGate Preprint, Dec 2024.

Ragab, M. I., et al., 2025. Multilingual Propaganda Detection using Transformer Models. Proc. of Nakba NLP, pp. 75–82.

Ribeiro, M. T., et al., 2016. "Why should I trust you?" Explaining predictions. Proc. of KDD, pp. 1135-1144.

Robert, E., et al., 2023. Comparative analysis of ML models for optimized SMS filtering. Journal of Cybersecurity Research.

Saha, S., and Nanda, A., 2023. BanglaNLP at BLP-2023: Benchmarking Transformer Models. Proc. of BLP, pp. 266–272.

Sarker, I. H., 2021. Cyberlearning: A new approach to cybersecurity education. IEEE Big Data, pp. 5550–5559.

Sarker, I. H., et al., 2020. SMS spam detection using machine learning: A survey. Journal of Big Data, 7(1), pp. 1–22.

Sasa Software, 2025. The AI Phishing Revolution: Implications for Cybersecurity in 2025. Published Online.

Shahriyar, M. F., and Tanbhir, G., 2025. Bangalabarta: A Benchmark Dataset for Smishing SMS Detection. Available at SSRN 5669250.

Simon, J., 2025. Advanced Deep Learning Models for Real-time Smishing Detection. Security and Privacy, 8(2), p. 102.

Tan, K. L., et al., 2022. RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis. IEEE Access, 10, pp. 21517-21525.

Tanbhir, G., et al., 2022. Hybrid Machine Learning Model for Detecting Bangla Smishing. Proc. of ICECEE, doi:10.1109/ICECEE64886.

Tanbhir, G., et al., 2024. Bangla SMS Dataset for Smishing Detection. IEEE Dataport. doi:10.21227/vxz9-ak04.

Tanbhir, G., et al., 2025. Hybrid Model for Detecting Bangla Smishing Text Using BERT. ResearchGate Preprint.

Uddin, M. S., et al., 2024. Machine learning based approach to categorize Bengali comments. PLOS ONE, 19(10), p. e0308862.

Vaswani, A., et al., 2017. Attention is all you need. Advances in Neural Information Processing Systems, pp. 5998–6008.

Zhao, Y., et al., 2025. Hybrid LSTM-Transformer Architecture with Multi-Scale Feature Fusion. MDPI Mathematics, 13(10), p. 1551.