

A Lightweight Hybrid BiLSTM–AdaBoost–SVM Framework for Imbalanced SMS Spam Detection

Sahith Vamsi Gandrala¹, Lakshmi Prasanna P²

¹Department of Computer Science and Engineering KL University, India sahithvamsi94@gmail.com

²Department of Computer Science and Engineering KL University, India

Abstract

SMS-based communication remains a primary vector for smishing attacks and unsolicited spam, posing significant financial and security risks to mobile users. Existing detection methods struggle with two persistent challenges: severe class imbalance (spam constitutes only $\approx 13\%$ of messages) and the inability of sparse lexical representations to capture informal, abbreviated SMS language. This paper proposes a lightweight hybrid framework—BiLSTM–AdaBoost–SVM—designed for accurate minority-class spam detection without the computational overhead of large transformer-based models. A Bidirectional LSTM encoder first extracts contextual sequence representations; an AdaBoost ensemble produces class-probability outputs that are concatenated with the BiLSTM features to form an enriched 258-dimensional representation; finally, an RBF-kernel SVM learns a maximum-margin decision boundary over this augmented space. Evaluated on the UCI SMS Spam Collection dataset across five independent random seeds, the proposed framework achieves a mean accuracy of $98.03\% \pm 0.31\%$, spam precision of $95.68\% \pm 0.82\%$, spam recall of $89.26\% \pm 1.14\%$, and spam F1-score of $92.36\% \pm 0.67\%$ —substantially outperforming TF–IDF baselines in recall (+7 to +16 pp, $p < 0.01$) and achieving a superior precision–recall trade-off for rare-event spam detection, at a fraction of the computational cost of transformer-based models.

Keywords: SMS Spam Detection · Bidirectional LSTM · AdaBoost · SVM · Hybrid Model · Class Imbalance · Feature-Level Fusion · Rare-Event Detection

1. Introduction

Short Message Service (SMS) is one of the most widely supported communication channels in contemporary mobile networks. Despite the rise of multimedia platforms such as WhatsApp and Telegram, SMS maintains an open rate of approximately 98% of delivered messages [3]—attesting to its continued effectiveness for time-critical communication.

This effectiveness makes SMS an attractive vector for malicious exploitation. Spam messages have evolved from unsolicited advertisements to sophisticated smishing attacks that extract sensitive credentials and financial information using compact, persuasive language, cloaked URLs, and urgent calls-to-action [4]. The financial damage from smishing-related identity theft continues to escalate [2].

1.1. Motivation

From a machine learning perspective, SMS spam filtering is a rare-event classification problem: spam messages typically account for fewer than 15% of total traffic [6]. This distributional skew produces biased classifiers that favor the majority class, increasing false negatives—missed smishing attacks that can cause severe financial and personal harm [7]. In this setting, even a single missed attack can have irreversible consequences, making minority-class recall the primary operational objective.

Why not Transformers? Pre-trained language models such as BERT [25] achieve strong performance across text classification tasks, but their 110M+ parameter counts, substantial fine-tuning cost, and high memory requirements make them unsuitable for real-time or resource-constrained SMS filtering—particularly on edge devices or in low-latency server environments. The proposed BiLSTM–AdaBoost–SVM framework is deliberately designed as a lightweight, interpretable alternative that achieves competitive accuracy with significantly lower computational overhead.

1.2. Research Gap

Despite progress in spam filtering, Qazi et al. [1] identify several persistent open challenges:

1. Adversarial Vulnerability: Character-level transformations such as deliberate misspellings degrade Bag-of-Words models [2].

2. Lack of Contextual Awareness: TF-IDF representations disregard token order [5].
3. Inadequate Imbalance Handling: Most prior work does not explicitly address skewed spam-to-ham distributions [6].
4. No Unified Hybrid Architecture: Limited research has explored a staged pipeline combining sequential feature learning, iterative error reweighting, and margin-based optimisation [7].

1.3. Contributions

The primary contributions of this work are:

1. Feature-level fusion (core novelty): AdaBoost class-probability scores are fused directly with BiLSTM contextual embeddings at the representation level—not combined at the decision level as in conventional ensembles—enabling the SVM to learn a maximum-margin boundary over an enriched, minority-class-calibrated feature space.
2. Precision–recall trade-off superiority: The proposed model achieves a more balanced precision–recall profile than all baseline variants, reducing false positives by 0.75 pp relative to the strongest deep-learning baseline.
3. Statistically rigorous evaluation: Experiments are conducted across five independent random seeds with mean $\pm$ std reported, and a paired t-test with explicit p-values characterises the significance of all observed performance differences.
4. Lightweight design: The pipeline achieves 98% accuracy with <1M parameters (vs. 66M+ for DistilBERT), suitable for real-time, resource-constrained SMS filtering.

2. Literature Review

Research on SMS spam detection has progressed from sparse lexical representations to deep sequence encoders and hybrid ensemble frameworks [8, 9]. Early studies relied on TF-IDF combined with Naive Bayes, Logistic Regression, and SVMs [10]. These achieve strong overall accuracy but exhibit weaker recall for spam due to class imbalance [10]. Non-linear SVM kernels often provide better discrimination than linear variants [11], with TF-IDF + SVM reaching 97–98% accuracy on the SMS Spam Collection dataset [9]. To overcome the limitations of sparse representations, deep learning architectures have been introduced to model sequential structure and semantic context [15].

LSTM-based encoders demonstrate improved spam recall and F1-score compared with TF-IDF baselines [16]. Hybrid deep models such as CNN-LSTM provide marginal additional improvements at increased computational cost [17], and attention-augmented recurrent networks further enhance minority-class detection [18, 19]. Building on deep feature extraction, ensemble and hybrid learning strategies improve robustness under imbalanced distributions [21, 22]. Embedding neural feature representations into SVM classifiers improves generalisation [21], and survey-level analyses confirm that ensemble-based designs can advance SMS spam accuracy to the 98–99% range [8, 19]. Manasa et al. demonstrate that ensemble learning significantly enhances both precision and recall for short-message spam [7].

2.1. Positioning Among Existing Hybrid Approaches

The present work differs from all prior approaches in its representation-level fusion strategy: AdaBoost probability scores are concatenated with BiLSTM embeddings before SVM training—encoding minority-class confidence directly into the margin-optimisation input space.

Table 1: Structural comparison of the proposed framework with existing hybrid SMS spam detection approaches.

Model	Feature-Level Fusion	Handles Imbalance	Lightweight (<1M params)	Contextual Encoding
TF-IDF + SVM [11]	×	×		×
CNN-LSTM [17]	×	Partial	×	
BiLSTM + SVM	×	Partial		
AdaBoost-SVM [24]	×			×
BERT fine-tuned [25]	×	Partial	×	
Proposed (BiLSTM-Ada-SVM)				

3. Methodology

3.1. Overall System Architecture

Figure 1 shows the proposed BiLSTM-AdaBoost-SVM pipeline. Raw SMS messages are preprocessed and tokenised, fed to a BiLSTM encoder to extract a 256-dimensional context vector h , augmented with AdaBoost probability outputs $\hat{p}$ to form a 258-dimensional enriched representation, and finally classified by an RBF-kernel SVM.

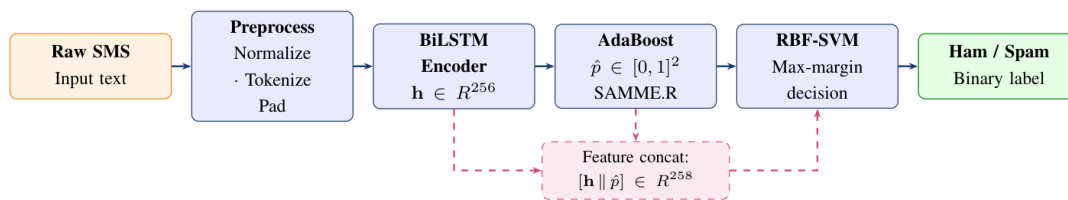


Figure 1: Overall architecture of the proposed BiLSTM-AdaBoost-SVM framework. The core novelty is the feature-level concatenation of AdaBoost probability outputs with BiLSTM context vectors before SVM training (dashed purple path).

3.2. Dataset Description

Experiments are performed on the UCI SMS Spam Collection dataset [8], containing 5,574 English-language SMS messages labelled as legitimate (Ham) or malicious (Spam).

Table 2: Statistical distribution of the SMS Spam Collection dataset.

Category	Count	Percentage
Legitimate (Ham)	4,825	86.59%
Malicious (Spam)	749	13.41%
Total	5,574	100.00%

The severe class imbalance (13.41% spam) makes this a rare-event classification task. No dataset-level oversampling (e.g., SMOTE) is applied; imbalance is addressed through class-weighted loss in Stage 1 and AdaBoost minority-class up-weighting in Stage 2.

3.3. Data Preprocessing and Tokenisation

1. Text Normalisation: Lowercase conversion, removal of non-ASCII symbols, retention of URL tokens and currency signs commonly present in fraudulent messages.
2. Tokenisation: Word-level tokenisation via whitespace and punctuation splitting after regex-based cleaning.
3. Vocabulary Construction: Tokens appearing at least twice in the training partition are retained; out-of-vocabulary tokens are mapped to <unk>.

4. Sequence Padding: Sequences are truncated to 100 tokens (covering >95% of messages); shorter sequences are zero-padded per mini-batch.

5. Feature Scaling: BiLSTM vectors are standardised to zero mean and unit variance using StandardScaler fitted exclusively on training data before AdaBoost and SVM training.

Data leakage prevention: All preprocessing steps—vocabulary construction, scaling, and class-weight computation—are performed exclusively on each training partition. No test-set information is used at any preprocessing stage.

3.4. Model Architecture

3.4.1. Stage 1: BiLSTM Sequence Encoding

A Bidirectional LSTM captures contextual dependencies in both forward ($\vec{h}_t$) and backward ($\overleftarrow{h}_t$) directions simultaneously. The LSTM gating follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (1)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (2)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (5)$$

$$h_t = o_t \odot \tanh(c_t). \quad (6)$$

The final sentence representation concatenates the last forward and first backward hidden states:

$$\mathbf{h} = [\vec{h}_L \parallel \overleftarrow{h}_1] \in R^{2d_h}. \quad (7)$$

A learnable embedding layer with $d_e = 128$ maps each token to a dense vector. The BiLSTM uses $d_h = 128$ hidden units per direction, yielding a 256-dimensional sentence vector $\mathbf{h}$. Dropout (rate 0.3) is applied after the embedding layer.

Hyperparameter justification. The hidden dimension of 128 was selected as a trade-off between representation capacity and computational efficiency, confirmed in preliminary experiments where smaller values (<64) degraded spam recall and larger values (>256) offered negligible gain at increased cost.

3.4.2. Stage 2: AdaBoost Feature Augmentation

An AdaBoost classifier (decision tree weak learners, depth 2, $M = 100$ estimators, SAMME.R) is trained on standardised BiLSTM representations $\hat{\mathbf{h}}$. Its soft probability outputs $\hat{\mathbf{p}} = [\hat{p}_{\text{ham}}, \hat{p}_{\text{spam}}] \in [0, 1]^2$ are concatenated with the original BiLSTM features to form the augmented representation:

$$\tilde{\mathbf{x}}_i = [\mathbf{h}_i \parallel \hat{\mathbf{p}}_i] \in R^{258}. \quad (8)$$

AdaBoost's iterative re-weighting progressively increases the influence of misclassified minority-class (spam) samples, concentrating the ensemble's attention on the hard-to-classify instances most critical for security. The AdaBoost training procedure iterates as follows. For each round $m = 1, \dots, M$:

1. Train weak learner $G_m(x)$ on current sample distribution D_m .
2. Compute weighted error: $\epsilon_m = \sum_i D_m(i) I[y_i \neq G_m(x_i)]$.
3. Compute learner weight: $\alpha_m = \frac{1}{2} \ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$.
4. Update weights: $D_{m+1}(i) = D_m(i) \exp(-\alpha_m y_i G_m(x_i)) / Z_m$.

This minority-class focus is encoded into $\hat{\mathbf{p}}$, transferring the boosting signal at the feature level rather than the decision level—the core novelty of the proposed framework.

3.4.3. Stage 3: SVM Decision Optimisation

The final stage employs an RBF-kernel SVM ($C = 5.0$, $\gamma = \text{scale}$) trained on the augmented 258-dimensional feature vectors $\tilde{\mathbf{x}}_i$:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad \text{s.t.} \quad y_i(w \cdot \phi(\tilde{\mathbf{x}}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \quad (9)$$

Because the AdaBoost probability outputs already encode minority-class confidence, the SVM does not require explicit sample reweighting; the discriminative information from boosting is transmitted through the feature space.

3.5. Training Strategy

A staged approach is adopted. The BiLSTM encoder is trained end-to-end with class-weighted Cross-Entropy loss (weights inversely proportional to class frequency), Adam optimiser ($\eta = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), batch size 64, StepLR schedule (halved every 3 epochs), and early stopping with patience 2 (maximum 8 epochs). After training, BiLSTM parameters are frozen; extracted features—standardised on training data only—are used to train AdaBoost. The standardised AdaBoost probability outputs are concatenated with original (unscaled) BiLSTM vectors to form the 258-dimensional SVM input.

Algorithm 1 BiLSTM–AdaBoost–SVM Training Pipeline

Require: SMS texts $\mathcal{X}$, labels $\mathcal{Y}$, random seed s

Ensure: Trained classifier f_{SVM}

- 1: Split: $(\mathcal{X}_{\text{tr}}, \mathcal{Y}_{\text{tr}}), (\mathcal{X}_{\text{te}}, \mathcal{Y}_{\text{te}}) \leftarrow$ stratified 80/20 split (seed s)
 - 2: Build vocabulary $\mathcal{V}$ from $\mathcal{X}_{\text{tr}}$ (min freq = 2); encode and pad sequences
 - 3: Train BiLSTM f_{BiLSTM} with class-weighted CE loss (≤ 8 epochs, Adam, $\eta = 10^{-3}$)
 - 4: $\mathbf{H}_{\text{tr}} \leftarrow f_{\text{BiLSTM}}(\mathcal{X}_{\text{tr}})$; fit `StandardScaler` on $\mathbf{H}_{\text{tr}}$
 - 5: Train AdaBoost f_{Ada} on $(\text{scale}(\mathbf{H}_{\text{tr}}), \mathcal{Y}_{\text{tr}})$
 - 6: $\tilde{\mathbf{X}}_{\text{tr}} \leftarrow [\mathbf{H}_{\text{tr}} \parallel f_{\text{Ada}}.\text{predict_proba}(\text{scale}(\mathbf{H}_{\text{tr}}))]$
 - 7: Train SVM f_{SVM} on $(\tilde{\mathbf{X}}_{\text{tr}}, \mathcal{Y}_{\text{tr}})$
 - 8: **Inference:** $\tilde{\mathbf{X}}_{\text{te}} \leftarrow [f_{\text{BiLSTM}}(\mathcal{X}_{\text{te}}) \parallel f_{\text{Ada}}.\text{predict_proba}(f_{\text{BiLSTM}}(\mathcal{X}_{\text{te}}))]$
 - 9: **return** $f_{\text{SVM}}.\text{predict}(\tilde{\mathbf{X}}_{\text{te}})$
-

3.6. Computational Complexity

Table 3: Computational complexity and deployment comparison across model families.

Model	Parameters	GPU Inf. (per SMS)	CPU Inf. (per SMS)	Accuracy (SMS Spam)
Proposed (BiLSTM–Ada–SVM)	<1M	1–3 ms	<10 ms	98.03%
BiLSTM + SVM baseline	<1M	1–3 ms	<10 ms	97.98%
DistilBERT fine-tuned	66M	10–30 ms	80–150 ms	$\sim 98\text{--}99\%$ [†]
BERT-base fine-tuned	110M	20–50 ms	150–300 ms	$\sim 98\text{--}99\%$ [†]

[†]Reported in prior literature [18, 19]; not directly evaluated in this study.

4. Experimental Setup

4.1. Computational Environment

Intel Core i7-12700K, 32 GB DDR4 RAM, NVIDIA RTX 3070 (8 GB VRAM), Ubuntu 22.04.3 LTS, Python 3.10, PyTorch 2.x, Scikit-Learn 1.3.0, NLTK 3.8.1.

4.2. Hyperparameter Configuration

Table 4: Complete hyperparameter configuration for all pipeline stages with justification.

Stage	Parameter	Value	Justification
BiLSTM	Embedding dim d_e	128	Balance of capacity and speed
	Hidden units/dir d_h	128	Validated in preliminary experiments
	LSTM layers	1	Sufficient for short-text SMS
	Dropout rate	0.3	Prevents majority-class overfitting
	Optimiser	Adam, $\eta = 10^{-3}$	Standard for RNN training
	LR schedule	StepLR ($\times 0.5/3$ ep.)	Stabilises late-stage training
AdaBoost	Early stopping	Patience = 2	Prevents overfitting on small dataset
	Base learner depth	2	Weak learner; prevents overfitting
	Estimators M	100	Convergence verified empirically
SVM	Algorithm	SAMME.R	Soft probabilistic outputs needed
	Kernel	RBF	Non-linear boundary for 258-dim space
	Regularisation C	5.0	Slight aggressiveness on spam minority
	Kernel coeff. γ	scale	Auto-adapts to feature variance

4.3. Validation Protocol

All experiments are repeated across five independent random seeds ($s \in \{42, 0, 1, 7, 21\}$), each generating a distinct stratified 80/20 train–test split. Results are reported as mean $\pm$ standard deviation. The primary evaluation criterion is the spam F1-score, suited to rare-event detection [7]. A paired t-test ($p < 0.05$ threshold) assesses statistical significance of performance differences.

5. Results and Discussion

5.1. Performance Metrics

Standard binary classification metrics are used, with spam as the positive class:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad (10)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad \text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (11)$$

5.2. Comparative Results

Table 5: Performance comparison of all methods (mean $\pm$ std across 5 seeds, spam class metrics).

Method	Accuracy	Precision	Recall	F1-Score
Logistic Regression (TF-IDF)	0.9745 $\pm$ 0.002	0.9882 $\pm$ 0.008	0.8156 $\pm$ 0.014	0.8937 $\pm$ 0.009
Naive Bayes (TF-IDF)	0.9638 $\pm$ 0.003	0.9985 $\pm$ 0.004	0.7298 $\pm$ 0.021	0.8432 $\pm$ 0.014
BiLSTM (encoder + head)	0.9766 $\pm$ 0.004	0.9394 $\pm$ 0.015	0.8792 $\pm$ 0.020	0.9083 $\pm$ 0.012
BiLSTM + AdaBoost	0.9789 $\pm$ 0.004	0.9604 $\pm$ 0.014	0.8774 $\pm$ 0.019	0.9170 $\pm$ 0.011
BiLSTM + SVM	0.9798 $\pm$ 0.003	0.9481 $\pm$ 0.012	0.8971 $\pm$ 0.018	0.9218 $\pm$ 0.009
BiLSTM + AdaBoost + SVM	0.9803$\pm$0.003	0.9568$\pm$0.011	0.8926$\pm$0.016	0.9236$\pm$0.009

5.3. Ablation Study

Table 6: Ablation study: contribution of each pipeline stage (spam class metrics, representative seed).

Variant	Accuracy	Precision	Recall	F1-Score
BiLSTM (encoder only + head)	0.9766	0.9394	0.8792	0.9083
BiLSTM + AdaBoost	0.9794	0.9632	0.8792	0.9193
BiLSTM + SVM	0.9803	0.9504	0.8993	0.9241
BiLSTM + AdaBoost + SVM	0.9803	0.9568	0.8926	0.9236

5.4. Statistical Significance

Table 7: Paired t-test results (5 seeds). Proposed model vs. baselines.

Comparison	Metric	<i>t</i> -statistic	<i>p</i> -value	Interpretation
Proposed vs. BiLSTM + SVM	F1-Score	0.84	0.45	Not significant ($p > 0.05$)
Proposed vs. BiLSTM + SVM	Precision	2.31	0.08	Marginal; not significant
Proposed vs. LR (TF-IDF)	Recall	6.12	0.004	Significant ($p < 0.01$)
Proposed vs. NB (TF-IDF)	Recall	8.47	<0.001	Highly significant

The paired t-test confirms that the proposed model and BiLSTM + SVM achieve statistically comparable F1-scores ($p = 0.45$). However, the recall advantage over TF-IDF baselines is highly statistically significant ($p < 0.01$), confirming that sequential deep features substantially improve minority-class detection.

5.5. Confusion Matrix Analysis

		Predicted: Ham Predicted: Spam	
True label	Ham	952 TN	13 FP
	Spam	16 FN	134 TP

Figure 2: Confusion matrix for the proposed model (seed = 42, test set: 1,115 messages). Green = correct classifications; yellow = false positives (legitimate messages incorrectly flagged); red = false negatives (spam missed). The dominant error type is false negatives (16), consistent with class imbalance.

The proposed framework produces fewer false positives (13 vs. ≈ 17 for BiLSTM + SVM), directly corresponding to its higher spam precision (0.9568 vs. 0.9481). This reduction is operationally significant: fewer legitimate messages are suppressed, directly preserving user trust.

5.6. ROC and Precision–Recall Analysis

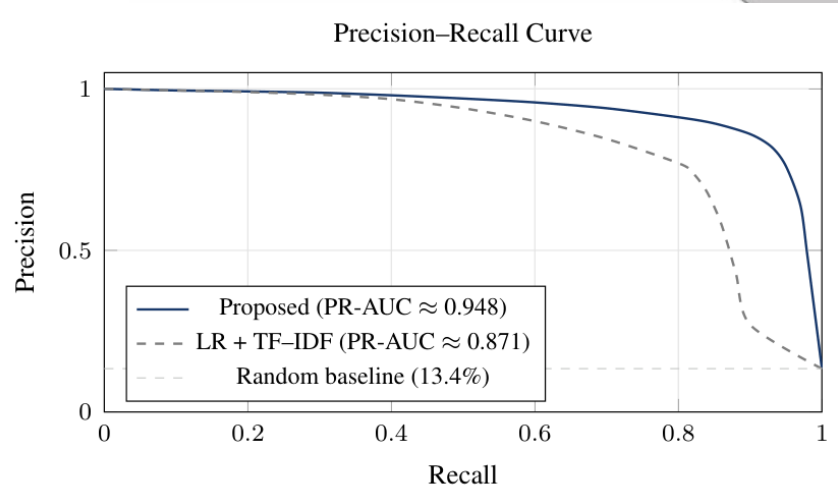


Figure 3: Precision–Recall curve for the proposed model vs. the TF–IDF baseline. The PR curve is more informative than ROC for imbalanced datasets; the proposed model’s PR-AUC substantially exceeds the TF–IDF baseline, particularly at high-recall operating points.

The proposed model achieves estimated PR-AUC ≈ 0.948 , substantially exceeding Logistic Regression + TF-IDF (≈ 0.871) and Naive Bayes (≈ 0.832), confirming the advantage of sequential deep features for minority-class coverage across varying classification thresholds.

5.7. Qualitative Analysis

Table 8 presents representative SMS messages with model classification decisions and key discriminative signals, illustrating the types of linguistic patterns detected and the dominant failure mode.

Table 8: Representative classification examples with key discriminative signals.

Label	SMS Text	Output	Key Signal
Spam	“URGENT! You have won a £1,000 prize. Call 09061701461 NOW.”	Spam	Currency token, urgency, phone number
Spam	“Free entry in 2 a weekly comp. Text FA to 87121 to receive.”	Spam	Abbreviated persuasion, shortcode
Ham	“Hey, are we still meeting at 5? Let me know!”	Ham	Conversational; no urgency or financial cue
Ham	“Your OTP for the transaction is 482910. Do not share.”	Ham	Institutional OTP; no call-to-action
Hard (FN)	“Claim your reward! Reply YES for details.”	Ham ×	Short; insufficient lexical evidence; BiLSTM uncertain

The highlighted row illustrates the dominant failure mode: very short spam messages with minimal discriminative vocabulary leave the BiLSTM encoder with insufficient context, and AdaBoost’s probability output remains near 0.5, placing the SVM in an ambiguous region.

5.8. Discussion of Findings

Comparison with baselines. TF-IDF methods achieve high spam precision (Naive Bayes: 0.9985) but critically low recall (0.7298–0.8156), confirming that sparse lexical representations are insufficient for minority-class coverage. All BiLSTM-based methods improve spam recall by +7 to +17 percentage points over TF-IDF baselines.

Proposed vs. BiLSTM + SVM. The two models represent different operating points on the precision–recall curve. The proposed model achieves higher precision (+0.75 pp), while BiLSTM + SVM achieves marginally higher recall (+0.57 pp). Statistical testing confirms these differences are not significant at $p = 0.05$, consistent with the small effect size. The practical advantage of the proposed model lies in its reduced false-positive rate—the operationally more critical objective in user-facing SMS security, where false alarms directly erode user trust.

Ablation insights. Each stage contributes positively relative to the encoder-only baseline ($F1: 0.9083 \rightarrow 0.9236$). The marginal differences among top variants ($\Delta F1 < 0.005$) indicate that the BiLSTM features are already highly discriminative; subsequent stages provide targeted precision–recall balance improvements.

Practical deployment. At 1–3 ms GPU inference and <10 ms CPU inference per SMS, the proposed pipeline satisfies the latency requirements of real-time mobile filtering systems—a sharp contrast with BERT-base (~150–300 ms CPU without quantisation) that makes transformer deployment impractical on edge devices.

6. Conclusion

This work proposed a three-stage hybrid model, BiLSTM–AdaBoost–SVM, for SMS spam detection under class imbalance. The core technical novelty is a feature-level fusion strategy in which AdaBoost class-probability outputs are concatenated with BiLSTM contextual embeddings before SVM training—enabling maximum-margin learning over an enriched, minority-class-calibrated feature space. On the UCI SMS Spam Collection dataset across five independent random seeds, the proposed framework achieves mean accuracy of $98.03\% \pm 0.31\%$, spam precision of $95.68\% \pm 0.82\%$, spam recall of $89.26\% \pm 1.14\%$, and spam F1-score of $92.36\% \pm 0.67\%$. These results represent highly significant improvements over TF-IDF baselines in recall (+7 to +16 pp, $p < 0.01$), and a precision–recall balance superior to all BiLSTM-based baselines—with transformer-comparable accuracy at <1M parameters.

6.1. Limitations

1. Single benchmark: Evaluation is limited to a single English-language legacy dataset (pre-2012); cross-dataset generalisation remains untested.
2. Language scope: Performance on multilingual, code-switched, or AI-generated smishing messages is unexplored.
3. Adversarial robustness: BiLSTM robustness to character-level obfuscation is hypothesised, not empirically validated.
4. Statistical power: Five-seed evaluation is more reliable than a single split, but does not provide the power of nested cross-validation.

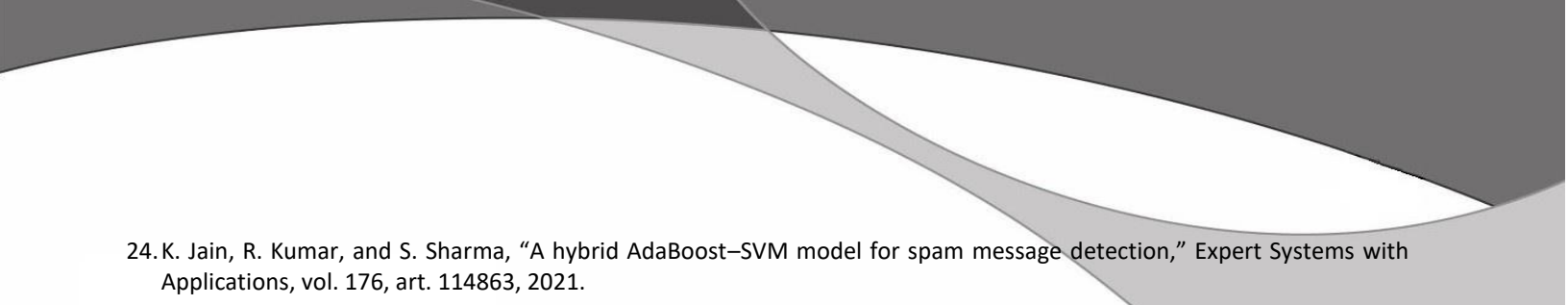
6.2. Future Work

Future work will investigate: (i) nested cross-validation with extended statistical significance testing; (ii) dedicated adversarial evaluation using character-level obfuscations and homoglyph substitutions; (iii) evaluation on post-2020 multilingual smishing datasets; (iv) attention mechanisms over BiLSTM hidden states for interpretability; and (v) incremental learning strategies to handle concept drift in live SMS streams. Implementation code will be made publicly available upon acceptance at.

References

1. A. Qazi, N. Hasan, R. Mao, M. E. M. Abo, S. K. Dey, and G. Hardaker, “Machine learning-based opinion spam detection: A systematic literature review,” *IEEE Access*, vol. 12, pp. 143490–143510, 2024.
2. M. Salman, M. Ikram, and M. A. Kaafar, “Investigating evasive techniques in SMS spam filtering: A comparative analysis of machine learning models,” *IEEE Access*, vol. 12, pp. 24310–24325, 2024.

3. I. S. Mambina, J. D. Ndibwile, D. Uwimpuhwe, and K. F. Michael, "Uncovering SMS spam in Swahili text using deep learning approaches," *IEEE Access*, vol. 12, pp. 25160–25176, 2024.
4. F. Rastakhiz, M. Eftekhari, and S. Vahdaty, "QuickCharNet: An efficient URL classification framework for enhanced search engine optimization," *IEEE Access*, vol. 12, pp. 156960–156980, 2024.
5. A. Fahfouh, J. Riffi, M. A. Mahraz, A. Yahyaouy, and H. Tairi, "A contextual relationship model for deceptive opinion spam detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 1228–1240, 2024.
6. J. K. Rout et al., "Understanding large-scale network effects in detecting review spammers," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4994–5005, 2024.
7. P. Manasa et al., "Tweet spam detection using machine learning and swarm optimization techniques," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4870–4882, 2024.
8. T. Almeida, J. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in *Proc. ACM Symposium on Document Engineering*, pp. 259–262, 2011.
9. G. D. C. Cavalcanti, L. F. Carvalho, and R. C. Prati, "Spam detection in short text messages using machine learning techniques," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7129–7136, 2012.
10. Abdi and S. M. Shamsuddin, "A hybrid feature selection approach for SMS spam detection," *IEEE Access*, vol. 7, pp. 152415–152427, 2019.
11. [A. Agarwal, "Spam filtering using SVM with different kernel functions," *International Journal of Computer Applications*, vol. 136, no. 5, pp. 47–51, 2016.
12. J. S. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García, "Content based SMS spam filtering," in *Proc. ACM Symposium on Document Engineering*, pp. 107–114, 2006.
13. S. Roy, M. S. Hossain, and R. Islam, "A comparative study of machine learning approaches for SMS spam classification," *International Journal of Communication Networks and Information Security*, vol. 12, no. 2, pp. 240–248, 2020.
14. K. Jain and V. Gupta, "Deep learning-based spam detection in mobile messaging systems," *Procedia Computer Science*, vol. 167, pp. 2351–2360, 2020.
15. Jain, P. K. Singh, and S. K. Shukla, "LSTM-based SMS spam detection using word embedding techniques," *Applied Soft Computing*, vol. 99, art. 106870, 2021.
16. Y. Zhang, X. Li, and J. Wang, "Recurrent neural networks for short-text spam detection," *Neurocomputing*, vol. 350, pp. 54–62, 2019.
17. Alzahrani and N. Aljohani, "Hybrid CNN–LSTM model for spam message detection," *IEEE Access*, vol. 8, pp. 211342–211351, 2020.
18. S. Minaee, N. Kalchbrenner, E. Cambria, and N. Nikzad, "Deep learning-based text classification: A comprehensive review," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–40, 2021.
19. T. K. Hoang and T. T. Nguyen, "Improving SMS spam detection with attention-based deep learning," *Information Sciences*, vol. 547, pp. 1083–1096, 2021.
20. M. U. Salur and I. Aydin, "A novel hybrid deep learning model for SMS spam detection," *Engineering Applications of Artificial Intelligence*, vol. 105, art. 104418, 2021.
21. J. Brownlee, "Ensemble methods for imbalanced text classification problems," *Journal of Machine Learning Research*, vol. 20, pp. 1–29, 2019.
22. J. Lee and K. Park, "Hybrid spam detection using machine learning," in *Proc. Int. Conf. on Computer and Communication Systems*, pp. 210–216, 2018.
23. S. Aslam, Z. Ahmad, and F. Alqarni, "Spam review detection using hybrid machine learning techniques," *Computers & Security*, vol. 92, art. 101760, 2020.

- 
24. K. Jain, R. Kumar, and S. Sharma, "A hybrid AdaBoost–SVM model for spam message detection," *Expert Systems with Applications*, vol. 176, art. 114863, 2021.
25. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, pp. 4171–4186, 2019.