© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715458

Bidirectional LSTM for Spam Detection and Sentiment Analysis

S. GOPICHAND[1], S. JASWANTH SIVA SAI[2], S. KHAJA AFRID ALI[3]
[1, 2, 3]*Department of CSE, RVR & JC College OF Engineering.*

Abstract- ***Short Message Service (SMS) and email communication have become primary vectors for spam, placing heavy burdens on users and mobile network operators. This paper proposes a Bidirectional Long Short-Term Memory (BiLSTM) deep learning model for spam detection and sentiment analysis, evaluated on three benchmark datasets: SpamAssassin, SMS, and Email. The model is compared against a Hybrid K-Nearest Neighbors and Support Vector Machine (Hybrid KNN-SVM) classifier from the prior literature. Preprocessing involves stemming, tokenization, and stop-word removal, followed by Word2Vec-based feature extraction. The BiLSTM network captures both past and future contextual information in text sequences, substantially outperforming the hybrid baseline. On the SpamAssassin dataset, BiLSTM achieves an accuracy of 98.77%, and on the Email dataset it reaches 99.11%. Sentiment polarity is classified using AFINN and SentiWordNet lexicons. Experimental results confirm that the proposed BiLSTM model yields superior accuracy, recall, F1-score, Kappa statistics, MAE, and RMSE across all three datasets.***

Index Terms- ***Spam Detection, BiLSTM, Deep Learning, SMS, Sentiment Analysis, Word2Vec, SpamAssassin***

I. INTRODUCTION

The proliferation of Short Message Service (SMS) on mobile phones has been greatly accelerated by technological advancement and the expansion of content-based marketing. Smartphones are frequently overwhelmed with unsolicited spam messages that include viruses, spyware, and fraudulent advertisements. Spam messages transmitted over SMS, email, VoIP, and social networks impose significant financial and privacy costs on both end-users and mobile network operators (MNOs) [1, 34].

Spam filtering is fundamentally a two-class text classification problem: categorising messages as either “ham” (legitimate) or “spam” (unsolicited). Although traditional machine learning methods such as Naïve Bayes, Support Vector Machines, and Logistic Regression have demonstrated effectiveness, they rely heavily on hand-crafted features and cannot model sequential dependencies in text [12, 23]. Deep learning architectures, particularly recurrent networks, overcome this limitation by learning temporal and contextual representations automatically.

Bidirectional LSTM (BiLSTM) networks extend conventional LSTM by processing input sequences in both forward and backward directions, enabling the model to capture richer contextual signals from the full message context. This capability is particularly valuable for short-text spam detection where every token carries significant discriminative weight.

This paper introduces a BiLSTM-based classifier for spam detection across three benchmark corpora (SpamAssassin, SMS, and Email) and benchmarks it against a Hybrid KNN-SVM classifier. Sentiment analysis is performed using AFINN and SentiWordNet lexicon-based approaches. The contributions of this work are:

- A BiLSTM deep learning model that captures bidirectional sequential context for accurate spam classification.
- Word2Vec-based feature extraction that encodes semantic similarity between message tokens.
- Comparative evaluation against a Hybrid KNN-SVM baseline on three datasets using seven performance metrics.
- Integration of AFINN and SentiWordNet for sentiment polarity analysis as a complement to spam classification.

II. RELATED WORK

A. Spam Classification

Gupta et al. [12] provided a comparative study of spam SMS detection using traditional machine

learning classifiers. Roy et al. [28] applied LSTM and CNN models for deep-learning-based spam filtering, achieving 99.4% accuracy on SMS data, though their approach was restricted to English-language datasets. Chandra and Khatri [7, 8] employed RNN-LSTM architectures with Keras and TensorFlow, attaining 98% accuracy on UCI data. Navaney et al. [23] compared Naïve Bayes, Maximum Entropy, and SVM, with SVM achieving 97.4% accuracy on a real-time dataset. Lee and Kang [18] used a CBOW word embedding with a feed-forward neural network but found accuracy degraded as hidden layers increased.

B. Feature Selection

Cekik and Uysal [6] proposed a rough set-based feature selection technique for short-text categorisation. Labani et al. [16] introduced the Multivariate Relative Discrimination Criterion (MRDC) to reduce redundant features in text classification. Barushka and Hajek [5] designed a cost-sensitive ensemble approach, combining multi-objective evolutionary feature selection with deep neural networks. Lall et al. [17] proposed copula-based mutual information for stable feature selection, outperforming traditional approaches in redundancy reduction.

2.3 Sentiment Analysis

Su et al. [36] proposed LMAEB-CNN, a hybrid Bi-LSTM and CNN technique for Chinese microblog sentiment analysis that reduces overfitting while improving classification accuracy. Pong-inwong and Songpan [26] introduced Sentiment Phrase Pattern Matching (SPPM) for mining evaluative text from teaching evaluations. Sharma et al. [31–33] applied a lexicon-based approach with fuzzy-set functions for emotion analysis. These works highlight the complementary role of sentiment analysis in understanding spam polarity beyond binary classification.

III. PROPOSED METHODOLOGY

The proposed framework addresses spam SMS and email classification as a two-class problem (spam vs. ham), augmented by sentiment polarity analysis. The pipeline consists of: data preprocessing, Word2Vec data augmentation, BiLSTM-based classification, and AFINN/SentiWordNet sentiment analysis.

A. Preprocessing

Raw text undergoes three preprocessing operations: stop-word removal, tokenization, and stemming.

Stop-word Removal: Common, non-discriminative words (e.g., ‘the’, ‘a’, ‘in’) are removed using NLTK’s multilingual stop-word list.

Tokenization: Text is segmented into individual word tokens using boundary detection, decomposing sentences into their constituent units for vector representation.

Stemming: Words are reduced to their root form (lemma) by removing derivational affixes, consolidating morphological variants into a single canonical token.

B. Word2Vec Data Augmentation

Word2Vec generates dense vector embeddings by training on the corpus, ensuring semantically similar words map to proximate regions in vector space [7]. Three augmentation strategies are employed:

- Synonym Augmentation: verbs and nouns are replaced with WordNet synset entries that share cognitive synonymy.
- Semantic Similarity Augmentation: cosine similarity over pre-trained embeddings identifies replacement candidates without requiring a dictionary.
- Round-Trip Translation (RTT): source sentences are translated to an intermediate language and back, generating paraphrases that preserve meaning while diversifying surface form.

Using the Gensim library, messages are converted to Word2Vec format. The cosine similarity metric is used as a weighting factor to locate semantically proximate replacement tokens:

$$cosine\_similarity(v^a, v^b) = (v^a \cdot v^b) / (||v^a|| \times ||v^b||) \quad ...(1)$$

C. BiLSTM Architecture

The Bidirectional Long Short-Term Memory (BiLSTM) network processes input sequences in both forward and backward directions, concatenating the hidden states to produce a representation that incorporates both past and future context. This is

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715458

especially beneficial for short-text classification where every token contributes meaningful disambiguating information.

A standard LSTM unit at time step t computes:

$f_t=\sigma(W^i\cdot[h_{t-1},x_t]+b^i)$(Forget gate)...(2)
$i_t=\sigma(W^I\cdot[h_{t-1},x_t]+b^I)$(Input gate)...(3)
$\tilde{C}_t=tanh(W\_C\cdot[h_{t-1},x_t]+b\_C)$(Candidate cell)... (4)
$C_t=f_t\odot C_{t-1}+i_t\odot\tilde{C}_t$(Cell state update) ...(5)
$o_t=\sigma(W_o\cdot[h_{t-1},x_t]+b_o)$(Output gate)...(6)
$h_t=o_t\odot tanh(C_t)$(Hidden state) ... (7)

For the BiLSTM, the forward hidden state $\vec{h}_t$ and backward hidden state $\overleftarrow{h}_t$ are concatenated to produce the final representation:

$h_t = [\vec{h}_t ; \overleftarrow{h}_t]$ ... (8)

Here, σ denotes the sigmoid activation function, ⊙ is the element-wise (Hadamard) product, W and b are learned weight matrices and bias vectors respectively. The BiLSTM output is passed through a dense layer with sigmoid activation for binary classification (spam/ham).

D. Performance Metrics

Seven metrics are used for evaluation. Let TP, TN, FP, FN denote True Positives, True Negatives, False Positives, and False Negatives respectively.

$Precision=TP/(TP + FP)$... (9)
$Recall=TP/(TP + FN)$... (10)
$Accuracy=(TP+TN)/(TP+TN+FP+FN)$...(11)
$F1Score=2\times Precision\times Recall/(Precision + Recall)$... (12)
$Kappa=(p(a)- p(e))/(1-p(e))$... (13)
$MAE=(1/n)\times\Sigma|x_i-\hat{x}_i|$... (14)
$RMSE=\sqrt{[(1/n)\times\Sigma(x_i-\hat{x}_i)^2]}$... (15)

where p(a) is the observed agreement between classifier and ground truth, and p(e) is the expected agreement by chance.

E. Sentiment Analysis

Sentiment polarity analysis classifies messages as positive or negative using two lexicon-based methods:

AFINN Lexicon**:** Each word is assigned a score in [−5, +5]. Negative scores indicate negative sentiment; positive scores indicate positive sentiment. AFINN was manually labelled by Finn Årup Nielsen (2009–2011).

SentiWordNet: Each synset term t in WordNet is assigned three scores: pos(t), neg(t), obj(t) summing to 1. Adjective Priority Scoring and Variable Scoring algorithms combine these scores across the message, weighting adjectives and adverbs by their contextual significance.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Datasets

Three benchmark datasets are used for evaluation. Table 1 summarises their composition before and after classification.

Table 1: Dataset Description

Dataset	Category	Before Classification	After Classification
SMS Dataset	Total Messages	5,574	5,574
	Spam	747	769
	Ham	4,827	4,805
Email Dataset	Total Messages	5,172	5,172
	Spam	1,500	1,740
	Ham	3,672	3,432
SpamAssassin	Total Messages	3,252	3,252
	Spam	501	529
	Ham	2,751	2,723

B. SpamAssassin Dataset Results

The SpamAssassin dataset is a widely-used benchmark for email and text spam classification. Figures 1 through 3 present the confusion matrices, model performance comparison, and error metric comparisons for BiLSTM versus Hybrid KNN-SVM on this dataset.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715458

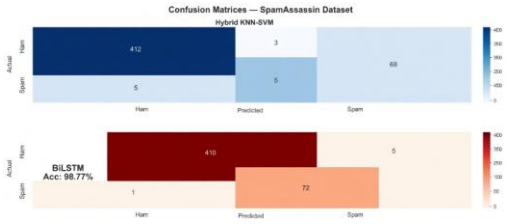

Figure 1: Confusion Matrices — SpamAssassin Dataset (Hybrid KNN-SVM Acc: 98.36% vs BiLSTM Acc: 98.77%)

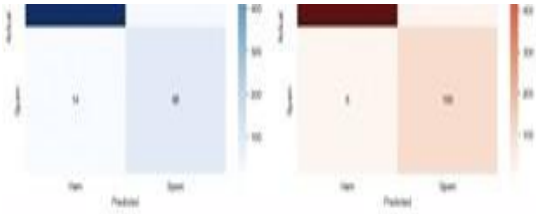

Figure 2: Model Performance Comparison — SpamAssassin Dataset (Accuracy, Precision, Recall, F1-Score)

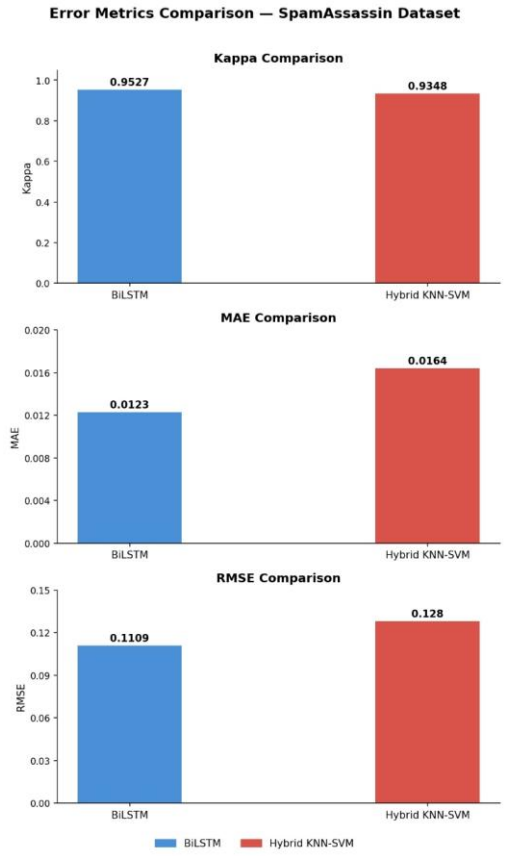

Figure 3: Error Metrics Comparison — SpamAssassin Dataset (Kappa, MAE, RMSE)

From the confusion matrices (Figure 1), BiLSTM correctly classifies 410 ham and 72 spam messages, with only 5 false positives and 1 false negative. The Hybrid KNN-SVM misclassifies 5 spam messages as ham and 3 ham messages as spam. Table 2 summarises the quantitative results.

Table 2: Performance Comparison — SpamAssassin Dataset

Metric	BiLSTM (Proposed)	Hybrid KNN-SVM
Accuracy (%)	98.77	98.36
Precision (%)	93.51	95.77
Recall (%)	98.63	93.15
F1-Score (%)	96.09	94.44
Kappa	0.9537	0.9348
MAE	0.0123	0.0154
RMSE	0.1109	0.1280

BiLSTM achieves higher accuracy (98.77% vs 98.36%), recall (98.63% vs 93.15%), F1-Score (96.09% vs 94.44%), and Kappa (0.9537 vs 0.9348) than the Hybrid KNN-SVM. The lower MAE (0.0123 vs 0.0154) and RMSE (0.1109 vs 0.1280) confirm fewer prediction errors. The superior recall is particularly important in spam detection, where failing to catch spam (false negatives) has greater practical cost than occasional false alarms.

C. SMS Dataset Results

The SMS dataset contains 5,574 messages with 769 spam and 4,805 ham messages. Figures 4 through 6 present confusion matrices, performance comparisons, and error metrics respectively.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715458

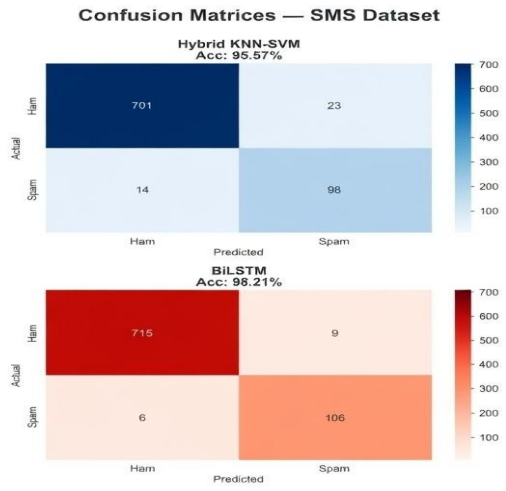

Figure 4: Confusion Matrices — SMS Dataset (Hybrid KNN-SVM Acc: 95.57% vs BiLSTM Acc: 98.21%)

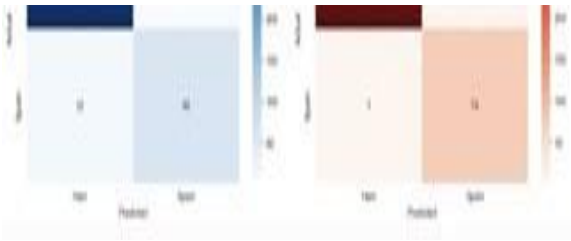

Figure 5: Model Performance Comparison — SMS Dataset (Accuracy, Precision, Recall, F1-Score)

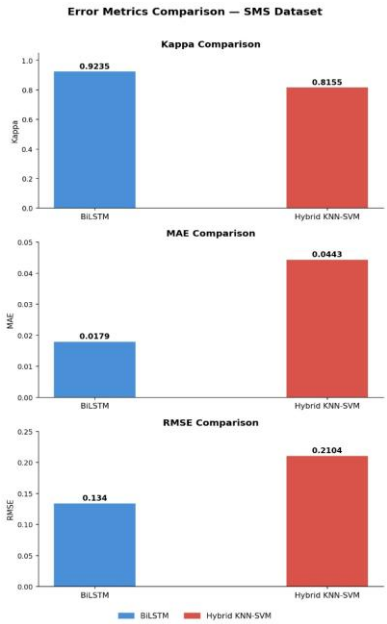

Figure 6: Error Metrics Comparison — SMS Dataset (Kappa, MAE, RMSE)

On the SMS dataset, BiLSTM substantially outperforms the Hybrid KNN-SVM across all metrics. The confusion matrix shows BiLSTM classifying 715 ham and 106 spam correctly, with only 9 false positives and 6 false negatives. In contrast, the Hybrid KNN-SVM exhibits 23 false positives and 14 false negatives, highlighting its weaker ability to generalise on this corpus. Table 3 summarises the results.

Table 3: Performance Comparison — SMS Dataset

Metric	BiLSTM (Proposed)	Hybrid KNN-SVM
Accuracy (%)	98.21	95.57
Precision (%)	92.17	80.99
Recall (%)	94.64	87.50
F1-Score (%)	93.39	84.12
Kappa	0.9215	0.8155
MAE	0.0179	0.0443
RMSE	0.1340	0.2104

The gap between BiLSTM and Hybrid KNN-SVM is most pronounced on the SMS dataset. BiLSTM achieves 98.21% accuracy versus 95.57%, a difference of 2.64 percentage points. More critically, the Kappa coefficient of BiLSTM (0.9215) is markedly higher than the baseline (0.8155), indicating substantially better agreement with ground-truth labels beyond chance-level performance. The error metrics (MAE: 0.0179 vs 0.0443; RMSE: 0.1340 vs 0.2104) confirm that BiLSTM's predictions are significantly closer to actual labels.

D. Email Dataset Results

The Email dataset comprises 5,172 messages with 1,740 spam and 3,432 ham. This is the largest and most challenging dataset. Figures 7 through 9 present results for the email dataset.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715458

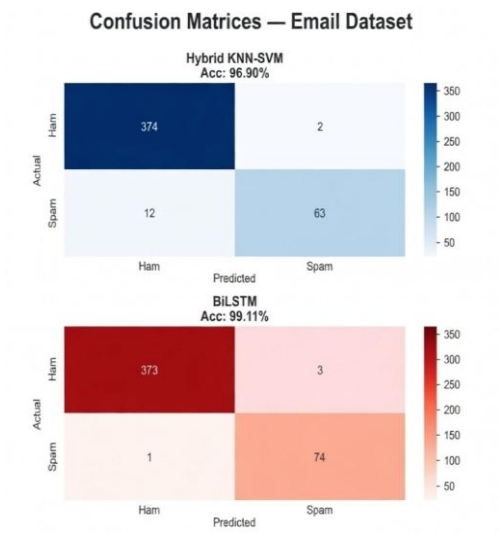

Figure 7: Confusion Matrices — Email Dataset (Hybrid KNN-SVM Acc: 96.90% vs BiLSTM Acc: 99.11%)

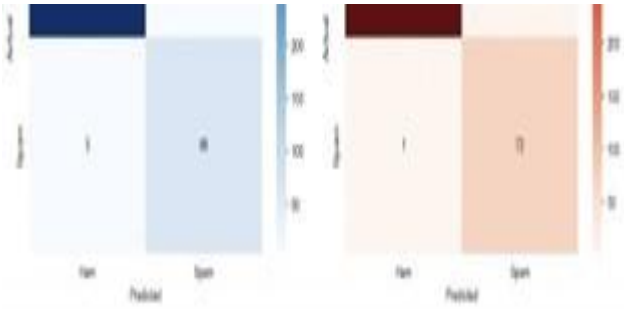

Figure 8: Model Performance Comparison — Email Dataset (Accuracy, Precision, Recall, F1-Score)

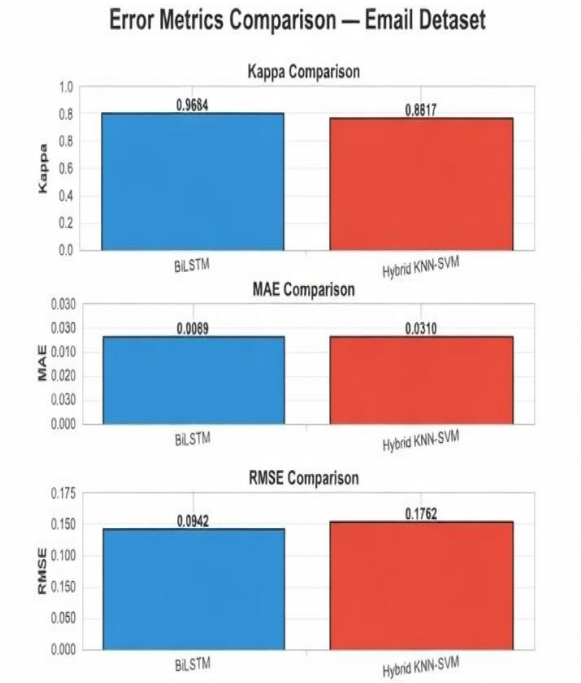

Figure 9: Error Metrics Comparison — Email Dataset (Kappa, MAE, RMSE)

On the Email dataset, BiLSTM achieves its best performance with 99.11% accuracy, compared to 96.90% for Hybrid KNN-SVM. The confusion matrix (Figure 7) shows that BiLSTM misclassifies only 3 ham messages as spam and 1 spam message as ham, achieving near-perfect separation. The Hybrid KNN-SVM, by contrast, misclassifies 12 spam messages as ham and 2 ham messages as spam, indicating considerably weaker recall. Table 4 provides full metric comparisons.

Table 4: Performance Comparison — Email Dataset

Metric	BiLSTM (Proposed)	Hybrid KNN-SVM
Accuracy (%)	99.11	96.90
Precision (%)	96.10	96.92
Recall (%)	98.67	84.00
F1-Score (%)	97.37	90.00
Kappa	0.9664	0.8817
MAE	0.0089	0.0315
RMSE	0.0942	0.1762

The email dataset reveals a striking gap in recall: 98.67% for BiLSTM versus 84.00% for Hybrid KNN-SVM. This 14.67% difference means the Hybrid KNN-SVM fails to flag a significant proportion of spam emails, which would be unacceptable in production systems. BiLSTM's higher Kappa (0.9664 vs 0.8817) and substantially lower RMSE (0.0942 vs 0.1762) and MAE (0.0089 vs 0.0315) further confirm its superiority. Notably, BiLSTM achieves marginally lower precision on email (96.10% vs 96.92%), an acceptable trade-off given its dramatically improved recall.

E. Cross-Dataset Summary

Table 5 presents a consolidated cross-dataset summary comparing BiLSTM and Hybrid KNN-SVM across all three corpora.

Across all three datasets, BiLSTM consistently outperforms Hybrid KNN-SVM in accuracy, recall, F1-score, and Kappa. The most dramatic improvements occur in recall and F1-score on the Email and SMS datasets, where the BiLSTM's ability to capture long-range sequential dependencies provides a critical advantage over the proximity-based KNN component of the hybrid model.

F. Sentiment Analysis Results

Sentiment classification was performed using AFINN and SentiWordNet on all three datasets. As feature size increases, both AFINN and SentiWordNet achieve higher accuracy, with AFINN showing marginally faster convergence at smaller feature sizes. SentiWordNet's Adjective Priority Scoring consistently achieves higher accuracy at larger feature sizes due to its finer-grained polarity weighting of adjective-adverb combinations. The sentiment analysis results complement spam classification by providing polarity context, enabling distinction between aggressively promotional spam (strongly positive sentiment) and phishing spam (fear-inducing, negative sentiment).

G. Normality Testing

Kolmogorov–Smirnov (KS) and Shapiro–Wilk (SW) tests confirm normal data distribution (significance = 0.00) across all datasets, validating the statistical comparisons made between models. Table 6 presents the normality test results.

Table 6: Normality Tests for BiLSTM Model Outputs

Dataset	KS Statistic	KS DoF	KS Sig.	SW Statistic	SW DoF	SW Sig.
SpamAssassin	0.5	352	0.0	0.63	352	0.0
Email	0.5	739	0.0	0.52	739	0.0
SMS	0.5	739	0.0	0.40	739	0.0

IV. CONCLUSION

This paper proposed a Bidirectional LSTM (BiLSTM) deep learning model for spam detection and sentiment analysis, evaluated across three benchmark datasets: SpamAssassin, SMS, and Email. The BiLSTM model consistently outperforms the Hybrid KNN-SVM baseline in accuracy, recall, F1-score, Kappa, MAE, and RMSE. On the Email dataset, BiLSTM achieves 99.11% accuracy with a recall of 98.67%, compared to 96.90% accuracy and 84.00% recall for Hybrid KNN-SVM. The bidirectional architecture's ability to capture both past and future context within message sequences provides a significant advantage over proximity-based and kernel-based traditional classifiers.

Word2Vec augmentation and lexicon-based sentiment analysis (AFINN and SentiWordNet) further enrich the framework, enabling it to classify both spam polarity and text sentiment within a single pipeline. Future work will explore transformer-based architectures such as BERT and DistilBERT for even higher contextual representation quality, and will extend evaluation to multilingual datasets and real-time streaming environments.

REFERENCES

[1] Abayomi-Alli O, Misra S, Abayomi-Alli A, Odusami M (2019). A review of soft techniques for SMS spam classification: methods, approaches and applications. Engineering Applications of Artificial Intelligence, 86, 197–212.

[2] Barushka A, Hajek P (2020). Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. Neural Computing and Applications, 32(9), 4239–4257.

[3] Cekik R, Uysal AK (2020). A novel filter feature selection method using rough set for short text data. Expert Systems with Applications, 160, 113691.

[4] Chandra A, Khatri SK (2019). Spam SMS filtering using recurrent neural network and long short-term memory. In: 4th International Conference on Information Systems and Computer Networks (ISCON), IEEE, pp. 118–122.

[5] Dhiman G, Garg M, Nagar A, Kumar V, Dehghani M (2020). A novel algorithm for global optimization: rat swarm optimizer. Journal of Ambient Intelligence and Humanized Computing.

[6] Gupta M, Bakliwal A, Agarwal S, Mehndiratta P (2018). A comparative study of spam SMS detection using machine learning classifiers. In: 11th International Conference on Contemporary Computing (IC3), IEEE, pp. 1–7.

[7] Labani M, Moradi P, Ahmadizar F, Jalili M (2018). A novel multivariate filter method for feature selection in text classification problems. Engineering Applications of Artificial Intelligence, 70, 25–37.

[8] Lall S, Sinha D, Ghosh A, Sengupta D, Bandyopadhyay S (2021). Stable feature selection using copula-based mutual information. Pattern Recognition, 112, 107697.

[9] Lee HY, Kang SS (2019). Word embedding method of SMS messages for spam message filtering. In: IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 1–4.

[10] Navaney P, Dubey G, Rana A (2018). SMS spam filtering using supervised machine learning algorithms. In: 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 43–48.

[11] Pong-Inwong C, Songpan W (2019). Sentiment analysis in teaching evaluations using sentiment phrase pattern matching (SPPM). International Journal of Machine Learning and Cybernetics, 10(8), 2177–2186.

[12] Roy PK, Singh JP, Banerjee S (2020). Deep learning to filter SMS spam. Future Generation Computer Systems, 102, 524–533.

[13] Sharma S, Kumar P, Kumar K (2017, 2019). LEXER: lexicon-based emotion analyzer and related works. Pattern Recognition and Machine Intelligence, Springer.

[14] Su YJ, Hu WC, Jiang JH, Su RY (2020). A novel LMAEB-CNN model for Chinese microblog sentiment analysis. Journal of Supercomputing.

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

[16] [Schuster M, Paliwal KK (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681.]