Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

SMS SPAM DETECTION USING HYBRID DEEP LEARNING

Zaid Khalaf Qasim
College of administration and Economics,
University of Misan, Amarah, Iraq
zaidallame944@uomisan.edu.iq

Zahraa Mousa Saad
College of administration and Economics,
University of Misan, Amarah, Iraq
Zahraa.m.s@uomisan.edu.iq

Abstract

The unprecedented increase of spam and promotional SMS messages has created the new challenge of quickly and accurately detecting these messages in the field of information security and communication. Users lose time and money to these spam messages, and these spam messages can also result in even greater financial and security risks. Therefore, building systems capable of detecting messages and analyzing their various features is extremely valuable.To combat the spam SMS phenomenon, this paper presents the first of its kind Hybrid Deep Learning Architecture, comprised of One-Dimensional Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM), and Transformer encoder blocks for SMS spam detection. The architecture is designed to accurately capture local features, temporal dependencies, and long-range relationships to classify messages into ham and spam. Preprocessing such as text cleaning and tokenization, padding, and balancing classes were done to the dataset in preparation for training the model. The model demonstrates substantial successful performance to all previous studies within the domain. Experimental results demonstrate that the proposed model achieves outstanding performance with 99.9% accuracy, 100% precision, 99.8% recall, 99.9% F1-score, and AUC = 1, outperforming previous methods. These findings highlight the potential of the

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

proposed hybrid architecture as an effective and practical solution for SMS filtering in real-world applications.

Keywords: SMS Spam Detection, Hybrid Deep Learning, Text Classification, Data Preprocessing

Introduction

As many people use mobile phones, and as the amount of messages sent daily rises, SMS spam has become a major issue in the cybersecurity and digital communications domains [1]. An SMS spam message can include advertisements, phishing attempts, malware, and can put people and organizations at great informational security risks, while also being heavily dissatisfying to end-users. In cases of large and rapidly growing data volumes, and in cases where methods of spam messages are growing in sophistication, the importance of rapidly and accurately detecting spam messages developed [2].

Traditional spam detection mechanisms, based on specific rules and/or keyword filters, typically struggle to identify new and variable sets of SMS messages [3]. The techniques have limited generalization, limited broad effectiveness, and deteriorate over time. In contrast, using machine learning and deep learning methods is better suited for detecting spam since these techniques automatically extract complex features and patterns from data. While analyzing the content of messages and the behavior of the senders, machine learning-adaptive approaches identify unauthorized messages with sufficient accuracy without the need for feature sets [4].

Given the importance of this issue, numerous studies have examined and developed various methods for SMS spam detection, which will be reviewed in the section along with an analysis of existing approaches.

Sethi et al. in [5] suggested using a deep learning technique for binary classification for spam detection in SMS. This used the Dense, LSTM, Bi-LSTM, and GRU models. The research was done on the popular UCL SMS dataset. The authors dealt with the class imbalance problem using undersampling, downsampling, and SMOTE methods. The Dense model gave the great accuracy and great results for spam detection in SMS.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Altunay and Albayrak in [6] developed a hybrid deep learning model that combines CNN and GRU for the purpose of detecting spam SMS in Turkih and English. This model underwent testing using the Turkih SMS Collection and UCI SMS Spam datasets. It was also compared with more conventional machine learning approaches. While it was determined that the CNN + GRU model was the most accurate with a propogated 99.07%, it is also recognized that the potential of this model is largely dependent on the extesive memory capacity available.

Ghourabi et al. in [7] proposed a hybrid model CNN-LSTM for the detection of SMS spam that can process messages written in Arabic and English. The authors of the model compared it with different traditional machine learning models and the results showed that the CNN-LSTM model achieved the highest performance for SMS spam and SMiShing attacks detection with an accuracy of 98.37%.

Abayomi-Alli et al. in [8] created a spam message detection system that implements a BILSTM model and assesses its performance on ExAIS_SMS and UCI datasets. Several classical machine learning algorithms were used as comparators and BiLSTM achieved superior performance with an accuracy of 93.4% on ExAIS_SMS and 98.6% on UCI.

Gomaa in [9] examined the efficacy of comparing seven deep neural networks and 5,574 SMS messages containing different spam. Comparing classic machine learning methods, deep learning approaches earned 99.26 accuracy through the RMDL architecture. Measuring the distance of machine spam, deep learning surpasses all predictors.

Altunay and Albayrak in [10] suggested a hybrid model combining CNN and GRU structures for SMS spam detection and assessed their model on the TurkishSMSCollection and UCI SMS Spam datasets containing Turkish and English messages. When considering traditional ML methods, the proposed model demonstrates 99.07% accuracy, surpassing other models. However, the authors report high memory usage as a limitation.

Kalyani et al. in [11] reviewed the use of RNN deep learning models for SMS spam detection on a Kaggle dataset containing 5,570 messages, using ADASYN to treat imbalanced classes and TF-IDF for feature representation. Several models were compared, RNN, LSTM, Bi-LSTM, and GRU, and the combined Bi-LSTM

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

+ GRU model proposed outperformed all others at an accuracy of 99% (increasing > 1% from the competing models) and outperformed both deep models standalone and a Random Forest baseline.

Al Saidat et al. in [12]. presented a hybrid CNN-BiLSTM model in that achieves SMS spam classification of Arabic text after cleaning, tokenization, padding, and other preprocessing steps. The model demonstrates accuracy of 96.99% and was trained with Adam optimizer using early stopping. The model also demonstrates high precision, recall, and F1-score, which proves that the model is effective in the Arabic text complexity.

Das et al. in [13] presented a method for SMS spam detection with a hybrid LSTM-DCNN model and Remora Optimization Algorithm (DL-ROA). This method is more adaptable to newly developed spam tactics and is not as reliant on frequency-based word encoding. Based on the results of the experiments conducted, this method exceeded the performance of current deep learning models as it accomplished an accuracy of 98.25%, as well as better recall and F1 score.

Sikarwar et al. in [14] did a comparative study of different models of deep learning for the detection of spam SMS. He and his team emphasized the dangers of phishing, fraud, and malware that spam SMS pose. Sikarwar et al. analyzed various models of deep learning. Their studies show that there is a consistent pattern of deep learning models outdoing classical models proving that deep learning models are robust and suitable for SMS spam filtering in real life.

Past research indicates that the use of CNNs and RNNs, which are also used in deep learning, demonstrates the ability of these models to pattern recognize beyond traditional rule-based or keyword-filtering methods. Although these models are able to analyze the content of message texts, there are still various issues such as low accuracy in detecting diverse spam messages, the inability to model long-term word dependencies, and the undersampling of significant textual features in many existing models.

While most studies have employed a single type of architecture, using multiple types of architecture increases feature variability, improving overall performance of the system. Moreover, many of the current deep learning models do not have mechanisms designed to focus on pertinent features adaptively. The addition of

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

the Transformer architecture blocks with attention mechanisms can enhance focus on the relevant information while suppressing noise, improving the performance of the model.

Within this framework, this paper a new integration of Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM) networks, and transformers designed specifically for SMS spam detection.

The main contributions of this research are as follows:

Hierarchical hybrid architecture: Utilizing CNNs for local feature extraction, BiLSTMs for modeling long-term dependencies, and Transformers for global attention across the entire text;

Simultaneous integration of local and global features: This combination allows the model to capture both short-term significant patterns and complex relationships between words;

Innovative design for complex text learning: The model's structure enables rich and comprehensive feature extraction from textual data without requiring manual modifications to the architecture or features.

The proposed framework, leveraging this hybrid architecture, offers strong capability in capturing diverse and key features from messages and provides an innovative tool for effective SMS spam detection.

The organization of the paper is as follows:

Section 2 introduces the fundamental concepts of Convolutional Neural Networks (CNNs), BiLSTMs, and Transformer blocks used in the proposed framework.

Section 3 presents the proposed methodology in detail, including data preprocessing, local and global feature extraction, attention-based mechanisms, and the final classification stage.

Section 4 describes the dataset used in the experiments.

Section 5 outlines the evaluation metrics employed to assess the performance of the proposed model.

Section 6 presents the simulation results and performance evaluation of the model in detecting various types of spam messages.

Section 7 provides a comparative analysis between the proposed framework and other state-of-the-art approaches.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Finally, Section 8 concludes the paper by discussing the findings and suggesting future research directions in SMS spam detection.

Basic concepts

In this section, a detailed overview of the fundamental principles and key concepts required to understand the proposed method is presented.

2.1 Convolutional Neural Network (CNN)

Convolutional Neural Network is a popular and crucial deep learning architecture that is designed for automatic feature extraction from structured data like images and signals as well as textual data. Compared to classical machine learning methods that involve significant manual feature engineering, CNNs can learn meaningful hierarchical features from data automatically [15].

CNNs rely on convolution operation and its mechanism. Fixed-size filters (kernels) move across the input and do element-wise multiplication to detect local patterns like edges, textures, and important sequences. This operation allows CNNs to learn local and hierarchical dependencies in the data [16].

A typical CNN architecture consists of several key layers [17]:

Convolution Layer: This layer is responsible for extracting initial features from the input. Each filter detects specific patterns, and the output is represented as a feature map.

Activation Layer: After convolution, nonlinear activation functions such as ReLU are usually applied to enhance the model's ability to learn complex relationships.

Pooling Layer: This layer reduces the dimensionality of feature maps, decreasing computational complexity and increasing robustness to noise. The most common type is Max Pooling.

Fully Connected Layer: In the final stages, the extracted features are fed into fully connected layers for final classification or decision-making.

In the field of text analysis and SMS spam detection, Convolutional Neural Networks (CNNs) are useful in identifying local patterns in the text, including n-grams, important keywords, and other notable phrases of the text. Incorporating recurrent networks like LSTM or GRU allows CNN models in analyzing and identifying not only local characteristics, but also the overall important contextual dependencies in the data [18].

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

In modern technology, CNN's impact is significant largely due to excellent precision, automated ability to learn features, and a diminished dependency on manual feature engineering. As a result, they have become highly proficient in spam detection, computer vision, and natural language processing.

2.2 Bidirectional Long Short-Term Memory

Bi-directional Long Short-term Memory (BiLSTM) is a form of recurrent neural network (RNN) systems designed for modeling long-term dependencies in sequential data. Long Short-term Memory (LSTM) networks were introduced to cover long-term remembering and gradient issues within standard RNN architecture. Three gates, as in LSTM architecture – Input Gate, Forget Gate and Output Gate – allow to store selective pieces and update in a long sequence of data [19].

BiLSTM is an advanced version of LSTM that processes sequential data in both directions:

Forward Layer: Processes the sequence from the beginning to the end.

Backward Layer: Processes the sequence from the end to the beginning.

By integrating the two layers in the network, the model has the capability of learning future and past dependencies all at once. This is an outstanding feature for NLP given how the meaning of a word is determined by the context on both sides of the word.

The main advantages of BiLSTM include:

Better learning of complex sequential patterns

Increased prediction accuracy for textual and time-series data

Suitability for tasks where bidirectional context matters, such as spam detection, machine translation, and sentiment analysis

2.3 Transformer

The Transformer is an advanced neural network design tailored to process sequential data; it works primarily in natural language processing (NLP), machine translation, and text analysis. RNNs and LSTMs process data sequentially, while Transformers analyze all data input simultaneously faster and more efficiently during training.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Transformers are based on an attention mechanism, predominantly self attention, which lets the model assess the value of one word with respect to other words in the sequence. This feature allows the model to identify long-term dependencies and relationships between tokens [20].

A typical Transformer model consists of two main components:

Encoder:

A stack of layers that processes the input tokens and extracts their semantic and structural features.

Decoder:

Uses the features extracted by the encoder to generate the final output, such as a translation or classification result.

Each layer in the Transformer includes:

Multi-Head Self-Attention: Enables the model to focus on different parts of the input simultaneously

Feed-Forward Network: Fully connected layers for further feature processing

Layer Normalization and Residual Connections: Help stabilize and improve learning

The main advantages of Transformers are:

Parallel processing of tokens, which significantly speeds up training

Better learning of long-term dependencies in text data

Improved performance on complex NLP tasks such as translation, sentiment analysis, and SMS spam detection

Methodology

In this research, we first design a deep, hybrid architecture that is able to perform feature extraction of the SMS text at the local level, while simultaneously learning long-range dependencies of the texts – an essential characteristic of natural language processing. To do so, we propose to integrate a one-dimensional Convolutional Neural Network (CNN) architecture, a bidirectional Long Short-Term Memory (BiLSTM) module, and a Transformer encoder block that are responsible, respectively, for recognizing local patterns, sustaining long-range dependencies, and applying global attention.This combination allows the model to simultaneously consider fine-grained textual features and broad semantic

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

relationships across different parts of the message, ultimately providing more accurate predictions of each message's class.

3.1 Data Preprocessing and Input Preparation

Deep learning models depend on the quality and standardization of input data. Noise, irrelevant information, and inconsistencies of entered details can result in lowered predictive accuracy. In the study, we read from a dataset containing ham and spam messages and the columns pertaining to message text and class labels are extracted. After extraction, the texts are subjected to a multi-step preprocessing pipeline, to ready them for the hybrid CNN + BiLSTM + Transformer model.

Text Cleaning

Every SMS message has to be converted to lowercase to avoid any learning discrepancies due to case differences when modelling certain features. Then, all URLs will be replaced with <URL> whereas all large numbers (3 digits and above) will be replaced with <NUM> so that large fragmented or uninformative data such as phone numbers and long strings of numbers will not negatively impact learning. Moreover, all text will be stripped of any and all non-alphanumeric characters, and superfluous whitespace will be reduced to better systematise and streamline the text. This stage serves to ensure that the inputs sent to the NLP model are smooth and regular. This eases the coming adjustments of the network weights and brings about rapid convergence of the model.

Message Length Distribution Analysis

To analyze message lengths before and after cleaning, and to better understand the data structure and choose a fixed token length, a histogram of message length was plotted. It shows that most SMSs and messages in the dataset, are quite short in length except for a few outliers which are long messages. This helps understand the maximum limit of token length that is optimal to cover most of the messages and minimizes paddings so that the model is able to run messages in batches uniformly.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Tokenization and Embedding

Once the data are cleaned, texts are transformed into numerical tokens utilizing a Tokenizer, which connects each word or token to an exclusive integer. All tokens are then made the same length, treating the shorter tokens through padding to maintain a constant sequence length across the data through the pad_sequences method.

After tokenization, each token is mapped to a dense vector in R^d space using an Embedding layer. These vectors preserve semantic information and similarities between words, enabling the model to learn both local and global semantic relationships in the text. The vector mapping formula is presented in Equation 1:

$$E_i = Embedding(x_i \in R^{max-len*d}) \quad (1)$$

where x_i is the SMS token and d is the embedding dimension.

Data Balancing

Within the SMS dataset, the ham class is predominant, and the spam class is lesser. Such an imbalance can lead the model to become biased to the majority class. To counter this, samples from the spam class are augmented using the Oversampling technique until the class sample sizes are equal. This ensures that the model learns the characteristics of the minority class.

Padding and Sequence Standardization

To have a uniform input length, all messages are padded to a maximum length, max_len, and longer messages are truncated to this length. This allows for batch training and concurrent processing of data, resulting in more stable weight updates and better convergence of the network.

Data Splitting

Once the dataset has been preprocessed and balanced, we split the data into three primary subsets: training, validation, and test. This split is done primarily to ensure the model can be evaluated fairly, and to avoid overfitting. With this in mind, the training data is 80% of the total data so the model can grasp the primary patterns. The other 20% is stratified (preserving the ham-to-spam ratio) and split

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

equally into the validation and test sets, with 10% for validation and 10% for testing.

The validation set assists with hyperparameter tuning, the selection of optimal weights, and the prevention of overfitting, while the test set is set aside until the end of the training for an assessment of the model's performance on completely new data. This three-way split of the data affords careful and accurate estimations of the utility of the model in development and guarantees the network learns both the ham and spam classes equitably.

3.2 Proposed Hybrid Network Architecture

Using one-dimensional Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM) networks, and Transformer Encoder blocks, we explore in this paper the integration of 3 supportive pieces of the architecture. This architecture will allow complexities of both short-range and long-range in SMS messages to be captured by the model.

Local Feature Extraction with CNN

The first processing part, in the suggested design, is a one-dimensional Convolutional Neural Network (CNN) which is applied directly to the sequence of vectors that the Embedding layer outputs. This part is focused to extract local features and short-range structures that are embedded in the SMS of a specific kind and are usually reflected in a small sequence of tokens. At this point, the input matrix, which is the filter of tokens, is processed by convolutional filters of fixed filter sizes. Each filter is created to discover a specific local element in the pattern and generates various feature maps by sliding over the text's temporal axis. The operation of the one-dimensional convolution layer is defined based on Equation 2:

$$f\left(\sum_{m=0}^{M-1} w_m^{(k)} \cdot E_{i+m,j} + b^{(k)}\right) = h_{i,j}^{(k)} \qquad (2)$$

In this equation, h_(i,j)^((k)) represents the output of the k-th filter at position i, E_(i+m,j) denotes the corresponding components from the embedding matrix, w_m^((k))are the filter weights, b^((k))is the bias term, and f(·) is the ReLU

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

activation function, which preserves nonlinearity and enhances feature discriminability.

When the convolution has finished computing the MaxPooling1D layer, the most important extracted features are consolidated by choosing the selecting the maximum values over certain time intervals, and the dimensionality of the data is reduced. Beyond allowing the features to be represented more simply, this also allows the model to be more robust to small variations of the sequence. The output of this component is more than just a representation of the local patterns in the text. It is a compressed and enriched representation, and this output is sent to the next stage of the network.

Sequential Dependency Modeling with BiLSTM

Once the CNN finishes extracting features locally, the highlighted and condensed representations from varying portions of the input text get forwarded to the BiLSTM layer to capture the short and long dependencies between the tokens both forward and reverse. The architecture of the layer, a bi-directional LSTM, allows the capture of forward (influences from previous tokens) and backward (influences from subsequent tokens) dependencies to occur simultaneously. The ability to do this is very useful when the input text is very short such as in the case of SMS messages because of the way word order and word placement can change the interpretation of the sentence.

Within the architecture of an LSTM network, the hidden state and the cell's state are updated and maintained by three separate gates, which are the input, the forget, and the output gates. The basic principles for the governing LSTM units are stated as follows

$$i_t = (w_i x_t + U_i h_{t-1} + b_i), \quad (3)$$

$$f_t = (w_f x_t + U_f h_{t-1} + b_f), \quad (4)$$

$$o_t = (w_o x_t + U_o h_{t-1} + b_o), \quad (5)$$

$$\tilde{c_t} = tanh(w_c x_t + U_c h_{t-1} + b_c), \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t}, \quad (7)$$

$$h_t = o_t \odot tanh(c_t), \quad (8)$$

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

x_t denotes the input at time t (in this context, the feature vectors produced by the CNN),
h_t represents the hidden state, c_t represents the cell state, and σ(·)is the sigmoid activation function.
In the bidirectional setting, the sequence is processed by two separate LSTM networks which work in opposite directions, and their outputs are combined at every time step:

$$h_t^{Bi} = [h_t^{forward}; h_t^{backward}] \quad (9)$$

Thus, the finishing output from the BiLSTM layer has the capability of incorporating a bidirectional construction of the representation of information flow, as it has the capabilities of encoding different semantic relationships, as well as some syntactic dependencies, and moderate long-term patterns among the tokens—even if the messages are short, informal, or lexically different.
The proposed architecture takes the unc compressed output from the BiLSTM and sends it directly to the Transformer module to bolster further refinement and contextualization of the sequence features via the global attention mechanism. As such, the BiLSTM layer serves as an analytical link between the CNN's local feature extraction and the Transformer's global semantic model, allowing the model to gain a fusion of low-level structural details and high-order contextual relations.
Modeling Long-Range Dependencies and Attention Mechanism with Transformer Encoder
After going through the BiLSTM output, containing a biderectional representation of the SMS sequencd along with the temporal dependencies among the tokens, these data are barreled into Transformer encoders to capture the long-range features and the semantic relationships among all tokens across the text messages. The Transformer module is implemented to better the model's attention towards important segments of the text and to capture intricate semantics that wouldn't be easily observable through simple local or sequential analyses.
Every Transformer Encoder block has a Multi-Head Attention component and a Feed-Forward Network component, which are joined together via a Residual Connection and Layer Normalization. This design guarantees that there will be

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

no vanishing or exploding gradients in the model while still keeping critical information from the tokens.

With the implementation of multi-head attention, the model is able to focus on various sections of the sequence during every step. This also helps the model to understand the long-range dependencies between non-adjacent tokens. The calculations of multi-head attention are carried out like this:

$$attention(Q, K, V) = softmax(\frac{QK^T}{d_k})V \quad (10)$$

where:

Q (Query) is a matrix representing the model's query or need to focus on different parts of the sequence.

K (Key) is a matrix representing the features of all tokens for computing similarity with Q.

V (Value) contains the actual values of each token, which are returned after attention weighting.

d_k is the key dimension, used for scaling before applying softmax to ensure stable gradients.

In the multi-head architecture, this operation is performed in parallel for h=4 different heads, and each head can learn independent and distinct dependencies among tokens. The outputs of the heads are then concatenated and mapped using the weight matrix W_o:

$$multiHead(Q, K, V) = concat(head_1 \ldots\ldots\ldots head_h)W_o \quad (11)$$

Utilizing several heads lets the model pay attention, at once, to each different aspect of the message's meaning. It also lets the model capture long-distance relationships among separagted tokens more effectively.

Following the attention path, we feed the network into a FNN, which is a composition of two Dense layers with ReLU activations, and a Dropout layer is added to avoid overfitting. Residual connections fused with layer normalization are used to keep the primary signals emanating from the tokens undisturbed, which results in stable training of the model. For each transformer block, the primary actions are as follows, First, the input x passes through a Multi-Head

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Attention layer. The output of this layer is then added to the original input x via a residual connection, followed by Layer Normalization:

$$x' = LayerNorm(x + multiheadAttention(x, x, x)) \quad (12)$$

Next, x′ passes through a Feed-Forward Network. The output of this network is again added to x′ via a residual connection and Layer Normalization is applied a second time:

$$x'' = LayerNorm(x' + feedforward(x')) \quad (13)$$

Here, x is the input to the Transformer block, x′ is the intermediate output after applying Multi-Head Attention and Layer Normalization, and x″ is the final output of the block after passing through the Feed-Forward Network and the second Layer Normalization.

Ultimately, the output from the Transformer blocks is kept as a whole sequence and is converted to a feature vector using GlobalAveragePooling1D. This is the representation of the SMS as it carries the local information learned by the CNN, the temporal dependencies captured by the BiLSTM, and the long temporal dependencies plus global attention thanks to the Transformer.

Only the Transformer Encoder can be implemented in this architecture as there is no need for a decoder. This is due to the task being a classification problem where the output is a single class being either spam or ham, which doesn't require a sequence to be generated. The Encoder is for feature extraction and long-range dependencies modeling. Employing Multi-Head Attention with a Feed-Forward Network, semantic relationships between the tokens are mapped.

At last, the aforementioned vector is linked to the Dense layer and, taking advantage of a sigmoid activation function, the likelihood of the message being categorized as spam or ham class is forecasted. The integration of CNN, BiLSTM, and Transformer Encoder into the hybrid structure provides the possibility of concurrent learning of local textual characteristics, sequential interdependence, and long-range semantic patterns, thereby enabling the model to effectively identify spam SMS messages with varying and brief patterns.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

At the end of this section, the block diagram of the proposed method is presented in Figure 1.

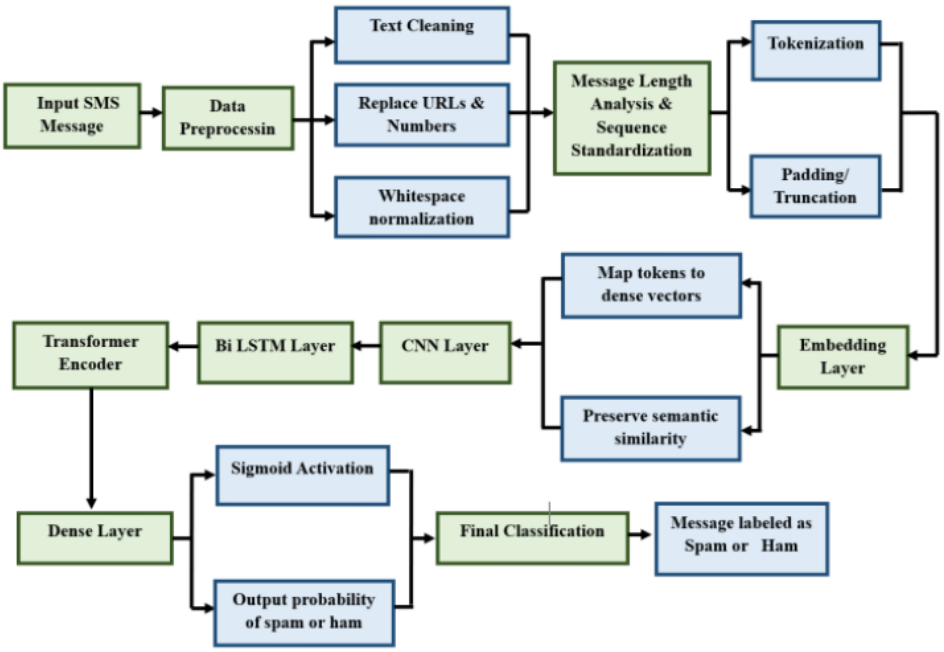

Figure 1. Flowchart of the proposed method

Dataset

For this project, we used a pre-existing dataset. This dataset is from Kaggle and is a common one used for SMS spam detection. It contains a collection of SMS messages and their corresponding labels indicating whether a message is spam or regular (ham). This dataset is in English, and we have it stored as a .txt document with two columns, the first one indicating the message's tags and the second one containing the message's body. In this dataset, there are two tags: the regular messages are tagged as ham, and spam messages are tagged as spam. These two tags are the basis of the binarized classification. There is enough data in the dataset to train and test the model effectively. What is most interesting is that there is a variety of messages including those with different lengths, structure, and phrasings.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Evaluation Metrics

To assess the performance of the model established in this study, the common metrics for binary classification problems have been used, which include Accuracy, Precision, Recall and F-1 score, which are defined as follows.

Accuracy

Accuracy refers to the proportion of correctly predicted instances in comparison to the total instances. Provided that all of the instances in question have been evaluated, this proportion gives a general view of the model's performance. However, this may not be the case in the presence of significant class distribution differences in the data.

The formula for calculating accuracy is:

$$Accuracy = \frac{TP_y+TN_y}{TP_y+TN_y+FP_y+FN_y} \qquad (14)$$

Where:

TP: True Positives (correctly predicted spam messages)

TN: True Negatives (correctly predicted non-spam messages)

FP: False Positives (non-spam messages incorrectly predicted as spam)

FN: False Negatives (spam messages incorrectly predicted as non-spam)

Precision

Precision reflects the percentage of messages correctly labeled as spam of all the messages that were predicted as spam. This is an important metric to use when evaluating the number of false positives (FP).

The formula for calculating precision is:

$$Precision = \frac{1}{n_c}\sum_y \quad (\frac{TP_y}{TP_y+FP_y}) \qquad (15)$$

Precision is crucial in scenarios where reducing false alarms is a priority.

Recall

Recall, also known as Sensitivity, refers to the portion of actual spam messages that were accurately recognized as spam. When it comes to false negatives (FN) and the cost associated with them, this metric becomes very important.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

The formula for calculating recall is:

$$Recall = \frac{1}{n_c}\sum_y \quad (\frac{TP_y}{TP_y+FN_y}) \tag{16}$$

F1-Score

The F1-score takes into account both Precision and Recall resulting in one measurement due to forming the harmonic mean and it takes on added significance when the necessity to find a compromise between Precision and Recall arises

Both precision and recall must be high for an F1-score to be high. Hence, it indicates the performance of the model holistically and, for cases that involve both errors, FP and FN, it is suitable to use it.

$$F_1 Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \tag{17}$$

Simulation results

This section details the output generated by running the proposed spam detection model, how well the model works in determining whether the message is ham/ spam is discussed. All the experiments were done in Python programming language, using the deep learning frameworks, Tensorflow and Keras. The model was run in a machine with Intel Core i7-13650HX processor, 16 GB of RAM, and a NVIDIA RTX 4060 GPU with 8 GB of GDDR6 VRAM.

Preprocessing Results

In this segment, the SMS data preprocessing stage including text cleaning, analyzing message lengths, tokenization, padding, and class balancing, has been completed. The text of the message was all converted to lowercase in order to assuage the model's learning concerns regarding uppercase and lowercase differences. Links were replaced with <URL> and any numbers with three or more digits were replaced with <NUM> indicators. Text was standardized and normalized when all non-alphanumeric characters were removed and extra spaces

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

were collapsed. This cleaning process helped stabilize the network weights, enabling faster model convergence. Before and after cleaning, message length analyses had the same shape, illustrated in the provided histogram charts (Figure 2); this showed that the majority of message lengths were short (5 to 30 words) and there were few relatively long messages. The length of the message distributions became somewhat more uniform and the length of messages were somewhat shortened after cleaning. As such, to minimize excessive padding while also covering all messages, a constant token length of 120 was determined.

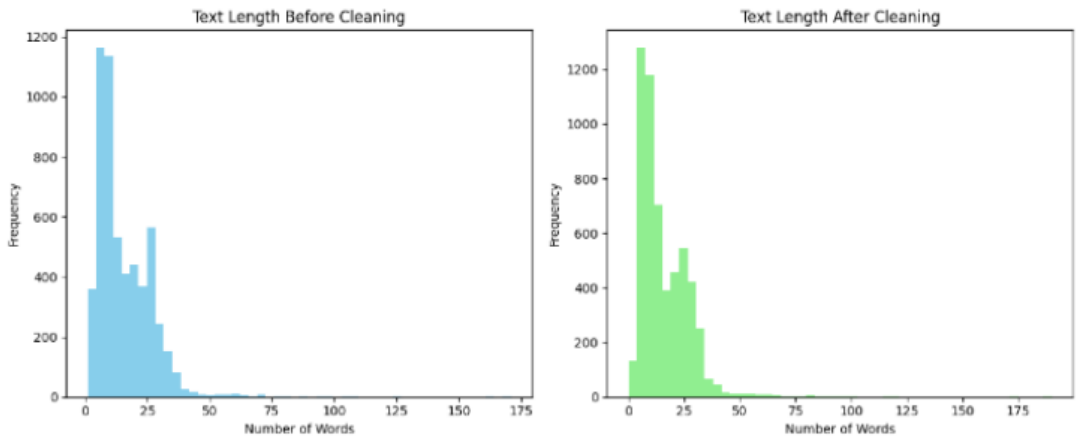

Figure 2: Histogram of SMS message lengths before and after text cleaning.

To analyze the lexical content of the dataset, the messages were first separated into ham and spam categories, and a WordCloud was generated for each group. In ham messages, the most frequent words consisted mainly of everyday expressions and conversational terms, including: i, you, to, the, a, u, and, in, me, my, is, it, that, of, for, s, have, can, so, and but (Figure 3, left). In contrast, spam messages predominantly contained vocabulary associated with advertisements, urgent promotional offers, and contact-related expressions, such as: to, a, call, you, your, free, the, 2, for, now, or, u, txt, is, on, ur, 4, have, from, and mobile (Figure 3, right).

Along with the Word Cloud visualization, the top 20 most common tokens in each category were calculated in bar chart form. For this analysis, there is an evident bottom-up approach demonstrating the lexical differences discriminating between

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

ham and spam messages along with the significant role of these common words in classification.

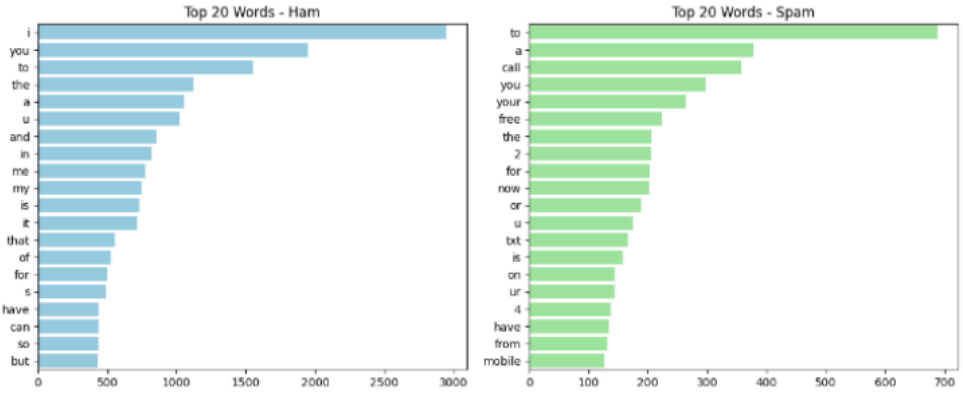

Figure3: Top 20 frequent words in ham (left) and spam (right) SMS categories

When analyzing the original dataset, it was observed that the class distribution was heavily imbalanced: around 14% of the messages were labeled spam, while around 86% were labeled ham. To mitigate this disparity, spam samples were duplicated to ensure that both classes were represented equally. One can clearly see through the bar charts before balancing (Figure 4) and after balancing that the two classes after oversampling achieved nearly the same sample size and thus reducing the chances of the model being biased to the class that had the majority.

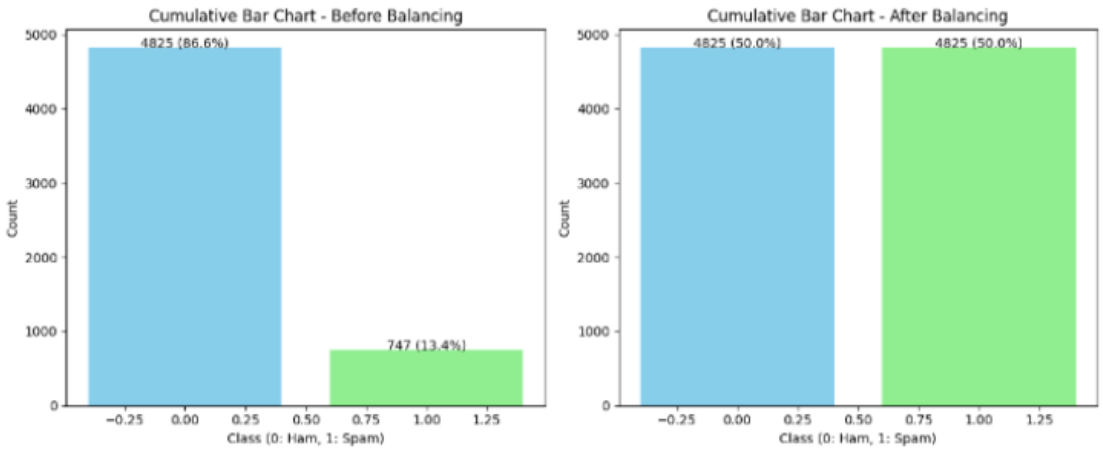

Figure4: Class distribution before and after applying oversampling

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Once the data had been cleaned and balanced, the text messages were converted to numerical tokens using the Tokenizer and then converted to a uniform input length of 120 through pad_sequences. This ensures stability and consistency while training and training while working with the text data in batches.

Overall, this stage in the preprocessing stage consisted of making the dataset consistent, uniform, and balanced. The lengths of the messages were normalized. The sequences were tokenized, prepared for embedding, and were ready for the hybrid-CNN + BiLSTM + Transformer model.

Model Architecture and Training Configuration

The input sequence consists of 120 tokens, mapped initially to a 128-dimensional vector space through an Embedding layer. An embedding dimension of 128 is a good trade-off to capture enough semantic information of words without growing computational expenses too much. The convolutional neural network consists of 160 convolutional filters with kernel size of 4, followed by a MaxPooling layer with a pool size of 2. Using 160 filters with a small kernel is good for obtaining short-term patterns and common n-grams in SMS texts, while MaxPooling helps to reduce dimensionality and focus on the most salient features.

The CNN output is then used as an input to an 80-unit BiLSTM layer in order to model temporal dependencies in both directions. Using 80 units seems to be an optimal choice as it allows the model to learn longer range dependencies in the messages without going to the higher side with computation. In order to maintain the same dimensions before going into the Transformer blocks, the features will be projected back to the embedding dimensions.

Two blocks of a transformer structure are stacked on top of one another. The structure is in a multi-head self-attention fashion, where we use four multipliers, as well as a feed-forward unit of size 256. Our choice of four heads means the model can capture various semantic dependencies. The feed-forward is of size 256 meaning we are working with a good balance of model size and computational cost. Furthermore, the use of residual connections, layer normalization, and dropout rates are set to 0.25 in order to enhance stability in training while reducing overfitting. The Global Average Pooling layer is then used to summarize the information across all the sequence after the transformer blocks.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

In this final part, one adds a dense layer with 128 neurons and ReLU, in which we select important merged features and, at the very end, have a single output neuron with a sigmoid activation for a binary classification (spam and not spam). The model makes use of an Adam optimizer intended for the binary cross-entropy loss with a 0.001 learning rate. This optimizer allows Adam to converge quickly and adapt its learning rate. A 0.001 learning rate seems to balance learning speed with the stability of the overall model. For this kind of classification problem, binary cross-entropy loss seems to be the most appropriate choice.

During training, recorded the accuracy and loss values for both the training and validation sets for each epoch. These logs are used to produce the training curves in Figure 5, which show how the hybrid model converged and how the validation performance stabilized due to Early Stopping.

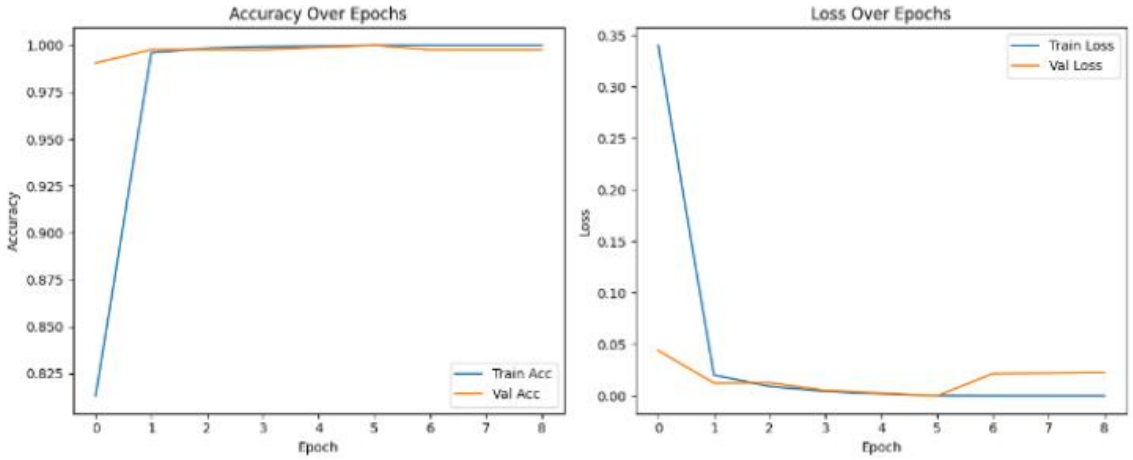

Figure 5:Training and validation accuracy and loss of the hybrid model.

Test Results and Performance Evaluation

This section discusses the outcome of the trained model concerning the test dataset. Evaluating the model will be done through Accuracy, Precision, Recall, F1 score, and AUC. A confusion matrix, ROC curve, and AUC will be further analyzed and presented.

The model is shown in confusion matrix in Figure 6. In this matrix, class 0 indicates non spam (ham) messages while class 1 indicates spam messages. The

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

confusion matrix shows the results on model predictions. The model managed to correctly classify 604 messages from the spam class to the non spam class (True Negatives) and 602 messages from the non spam class to the spam class (True Positives). As shown in the confusion matrix, the model only predicted one other class from the spam class and falsely predicted it as spam (False Positive), showing the model has an excellent ability to avoid false alarms.

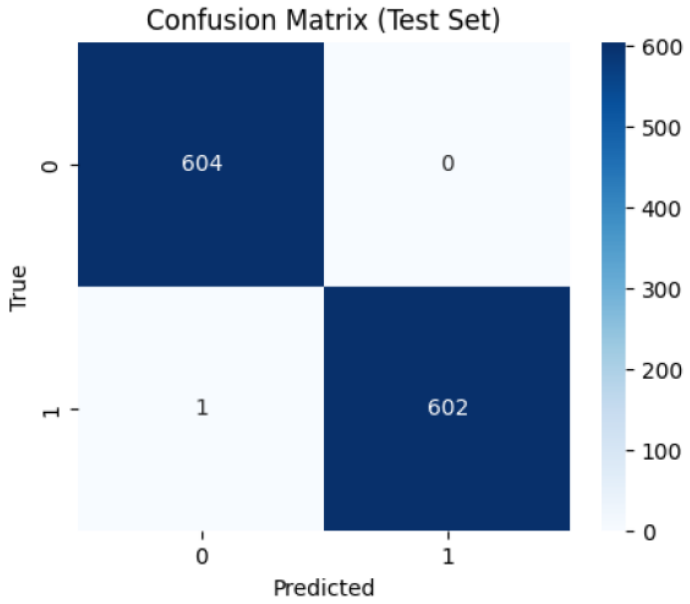

Figure 6: Confusion matrix of the test dataset

The model has a test ROC (Receiver Operating Characteristic) Curve (shown in Figure 7) to demonstrate its ability to discriminate between spam and non-spam . The horizontal axis (False Positive Rate) is the percentage of legitimate messages that the model incorrectly classifies as spam. The vertical axis (True Positive Rate) is the percentage of spam messages the model correctly classifies.

The performance of the model with an AUC value of 1 can be considered as a model performance showing a perfect result. That is, the model made no misclassifications and was able to distinguish between the spam and non-spam messages with no mistakes. This clearly shows the excellent performance of the hybrid model in effectively capturing the semantic and structural features of messages and predicting the correct class of each message.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

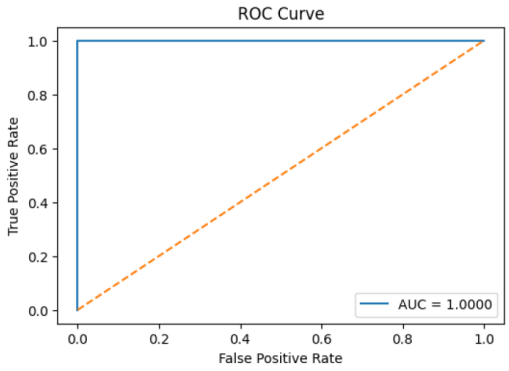

Figure 7: Receiver Operating Characteristic curve for test data

At the end of this section, Figure 8 presents the evaluation metrics obtained on the test set. The model achieved an Accuracy of 99.9%, Precision of 100%, Recall of 99.8%, and an F1-score of 99.9%.

These findings suggest that there was an outstanding success of the suggested hybrid model. The Perfect Precision shows that there were no false positives since all the messages that were predicted to be spam were actually spam. The high Recall shows that almost every actually spam messages were found and there was very little false negatives. As a result, for the F1 score, indicating high balance of Precision and Recall, there was a very high score, reflecting the overall great success of the model in detection of ham and spam messages. This strongly confirms that the model was a great success in their tasks for both classes.

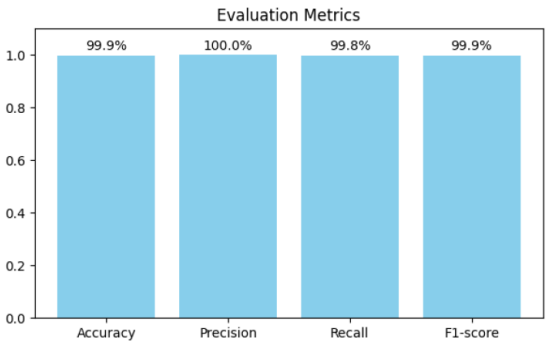

Figure 8: Performance metrics of the proposed model on the test data

Comparison

This section presents a training analysis of the alternative models used in the detection of SMS spam texts and illustrates the merits of the hybrid model through the demonstration of its effectiveness in the classification of spam and non-spam

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

messages along with its capacity to capture the semantic and structural attributes of messages. To guarantee an unbiased comparison, every technique utilized the same dataset.

Below, each method is briefly described, and comparative results are presented:

For SMS spam detection, the dataset from kaggle contains 5,570 entries. In the research [21], the Over sampling method ADASYN was used to counter the class imbalance, and TF-IDF embedding technique was used to capture the message content. A Random Forest model was first tested and got a 92.3% accuracy score. Then, multiple RNN architectures (Simple RNN, LSTM, Bi-LSTM, GRU) were tested. LSTM Models were the most successful and attained a score of 93% for RNN, 98% for LSTM, Bi-LSTM 98.2, and GRU 98.3. After all this, a hybrid model was introduced combining Bi-LSTM and GRU, which got 98.9 accuracy.

In the research attempt of [22], a method relying on multilingual embeddings was suggested for the detection of spam SMS. In the aforementioned research, AutoML multilingual embedding models were utilized to construct a machine learning model able to differentiate spam messages from ham messages. The experiment results show that the leading metrics for this model are 99 Accuracy, 98 Precision, 93 Recall, and 95 F1 score.

In the following, the results of the aforementioned methods are compared with the proposed hybrid model in this study. Table 1 summarizes the performance of each method in terms of accuracy.

Table 1: Comparison of SMS Spam Detection Techniques

Reference	Method	Accuracy (%)
[21]	Random Forest + TF-IDF	92.3
[21]	Simple RNN	93.0
[21]	LSTM	98.0
[21]	Bi-LSTM	98.2
[21]	GRU	98.3
[21]	Hybrid Bi-LSTM + GRU	98.9
[22]	Multilingual Embedding + AutoML	99.0
This Study	CNN + BiLSTM + Transformer (Proposed)	99.9

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

The hybrid model's accuracy is better than the accuracy of all previously reported methods, as shown in Table 1, this is due to several reasons. To begin with, the CNN layers extract local features and n-gram patterns that are indicative of spam messages. Following that, the model's ability to capture long-term dependencies in sequential data due to the BiLSTM component allows the model to better understand contextual relationships within messages. Additionally, the model's ability to capture long-range interactions and semantic dependencies across an entire message is due to the global attention mechanisms in the Transformer's blocks. The constructed architecture is better is selling the correct local and global representations of distinct features, which is why this approach and all other spam and ham message discrimination are false. It was the correct collection of patterns of the most important tokens at the structural and semantic relationships that this model was able to perform well.

Conclusion

This research focused on hybrid deep learning models with Convolutional Neural Networks, Bidirectional Long Short Term Memory, and Transformers, and evaluated their effectiveness in SMS spam classes. The models were able to learn both nearby token-level characteristics and long-distance semantic and structural dependencies in the text. The model was able to effectively learn the characteristics of spam and ham SMS after a series of preprocessing tokenization, padding, and balancing.

Experimental evaluation on the test set demonstrated outstanding performance: Accuracy = 99.9 %, Precision = 100 %, Recall = 99.8 %, F1 score = 99.9 %, and AUC = 1.0. These results confirm that the proposed method not only excels in classification accuracy, but also robustly captures meaningful textual patterns and minimizes both false positives and false negatives.

Additionally, when compared with other reported methods of detecting SMS spam, both traditional and deep learning, our hybrid model shows improvement and advantages of combining convolutional feature extraction with sequential modeling and global attention.

Due to its strong efficiency and generalization capabilities, the proposed model will be an efficient and reliable solution for operational spam filtering systems.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

Potential future endeavors could be directed toward expanding the approach's implementation to multilingual data sets and also examining adversarial robustness, as well as the potential for real-time deployment concerning large-volume SMS traffic.

References

[1] Liu, X., Lu, H., & Nayak, A. (2021). A spam transformer model for SMS spam detection. IEEE Access, 9, 80253-80263.
[2] Lota, L. N., & Hossain, B. M. (2017). A systematic literature review on sms spam detection techniques. International Journal of Information Technology and Computer Science (IJITCS), 9(7), 42-50.
[3] Xia, T., & Chen, X. (2020). A discrete hidden Markov model for SMS spam detection. Applied Sciences, 10(14), 5011.
[4] Abdulhamid, S. I. M., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., & Herawan, T. (2017). A review on mobile SMS spam filtering techniques. IEEE Access, 5, 15650-15666.
[5] Sethi, M., Tyagi, N., Kalsi, P. S., & Rao, P. A. (2023, May). Deep Learning-based Binary Classification for Spam Detection in SMS Data: Addressing Imbalanced Data with Sampling Techniques. In 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (pp. 1-9). IEEE.
[6] Altunay, H. C., & Albayrak, Z. (2024). SMS Spam Detection System Based on Deep Learning Architectures for Turkish and English Messages. Applied Sciences, 14(24), 11804.
[7] Ghourabi, A., Mahmood, M. A., & Alzubi, Q. M. (2020). A hybrid CNN-LSTM model for SMS spam detection in Arabic and English messages. Future Internet, 12(9), 156
[8] Abayomi-Alli, O., Misra, S., & Abayomi-Alli, A. (2022). A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset. Concurrency and Computation: Practice and Experience, 34(17), e6989

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

[9] Gomaa, W. H. (2020). The impact of deep learning techniques on SMS spam filtering. International Journal of Advanced Computer Science and Applications, 11(1).
[10] Altunay, H. C., & Albayrak, Z. (2024). SMS Spam Detection System Based on Deep Learning Architectures for Turkish and English Messages. Applied Sciences, 14(24), 11804.
[11] Kalyani, V. V., Sundari, M. R., Neelima, S., Prasad, P. S. S., Mohan, P. P., & Lakshmanarao, A. (2024, April). SMS Spam Detection using NLP and Deep Learning Recurrent Neural Network Variants. In 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS) (pp. 92-96). IEEE
[12] Al Saidat, M. R., Yerima, S. Y., & Shaalan, K. (2024). A novel approach for Arabic SMS spam detection using hybrid deep learning techniques. Procedia Computer Science, 244, 260-267
[13] Das, L., Ahuja, L., & Pandey, A. (2024). A novel deep learning model-based optimization algorithm for text message spam detection: L Das et al. The Journal of Supercomputing, 80(12), 17823-17848.
[14] Sikarwar, S. S., Arivukkodi, R., Krishnane, D., Sharma, H., Namdeo, A., & Jadon, K. (2024, December). Enhancing SMS Spam Detection Using Deep Learning Models: A Comparative Study. In 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N) (pp. 1367-1372). IEEE.
[15] Ajit, A., Acharya, K., & Samanta, A. (2020, February). A review of convolutional neural networks. In 2020 international conference on emerging trends in information technology and engineering (ic-ETITE) (pp. 1-5). IEEE.
[16] Ketkar, N., & Moolayil, J. (2021). Convolutional neural networks. In Deep learning with Python: learn best practices of deep learning models with PyTorch (pp. 197-242). Berkeley, CA: Apress.
[17] Krichen, M. (2023). Convolutional neural networks: A survey. Computers, 12(8), 151.
[18] Zhang, X., Zhang, X., & Wang, W. (2023). Convolutional neural network. In Intelligent Information Processing with Matlab (pp. 39-71). Singapore: Springer Nature Singapore.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 01, Issue 09, December, 2025
Website: usajournals.org

[19] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016, August). Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers) (pp. 207-212).

[20] Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. Advances in neural information processing systems, 34, 15908-15919.

[21] Kalyani, V. V., Sundari, M. R., Neelima, S., Prasad, P. S. S., Mohan, P. P., & Lakshmanarao, A. (2024, April). SMS Spam Detection using NLP and Deep Learning Recurrent Neural Network Variants. In 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS) (pp. 92-96). IEEE.

[22]Sachan, R. K., Tiwari, V., Raghav, N., & Rajpoot, P. (2025). Spam Text Messages Detection Using Multilingual Embedding. Procedia Computer Science, 260, 390-398.