SMS Spam Detection Using BERT and Multi-Graph Convolutional Networks

Linjie Shen[a], Yanbin Wang[b], Zhao Li[c], Wenrui Ma[d,e]

[a]*School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, , Zhejiang, China*
[b]*Department of Engineering, Shenzhen MSU-BIT University, , Shenzhen, 518172, Guangdong, China*
[c]*Zhejiang Lab, , Hangzhou, , Zhejiang, China*
[d]*School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, , Zhejiang, China*
[e]*Department of Big Data and Future E-Commerce Technology, Zhejiang Key Laboratory, Hangzhou, , Zhejiang, China*

Abstract

The surge in smartphone usage has significantly increased SMS traffic and, consequently, SMS spam, posing risks such as phishing, financial losses, and privacy breaches. Traditional rule-based and blacklist methods fail against evolving spamming techniques, prompting the adoption of machine learning and deep learning approaches. However, models like CNNs and RNNs struggle to capture global co-occurrence patterns and complex semantics, while transformer-based models like BERT lack explicit syntactic and co-occurrence modeling. To address these limitations, we propose the BERT with Triple-Graph Convolutional Networks (BERT-G3CN) model, the first framework to integrate BERT word embeddings with graph embeddings from Cooccurrence, Heterogeneous, and Integrated Syntactic Graphs. This multi-graph approach captures diverse features and models both global and local structures using tailored Graph Convolutional Networks (GCNs). Experiments on two benchmark datasets demonstrate that BERT-G3CN achieves superior accuracy of 99.28% and 93.78%.

Keywords: SMS spam detection, BERT, Graph Convolutional Networks

1. Introduction

The proliferation of smartphones has revolutionized the way people communicate [1], leading to an unprecedented surge in Short Message Service (SMS) usage worldwide. With billions of SMS messages sent daily, this communication medium has become integral to both personal and business interactions [2]. However, the massive increase in SMS traffic has also attracted the attention of malicious actors, resulting in a significant rise in SMS spam [3]. These unsolicited messages, which often contain phishing links, fraudulent offers, and other forms of deception, pose substantial risks to individuals and organizations alike.

The impact of SMS spam extends far beyond mere annoyance. Phishing attacks embedded in these messages can lead to severe financial losses, unauthorized access to sensitive information, and widespread privacy breaches [4]. For businesses, the consequences can be even more dire, including damage to brand reputation, legal liabilities, and compromised customer trust. As the tactics employed by spammers become increasingly sophisticated, traditional methods of filtering and blocking spam are often insufficient, leaving users vulnerable to the ever-evolving threat landscape [4, 5]. The sheer volume of SMS spam has created a pressing need for effective detection and mitigation strategies [6]. Unlike email spam, which has been the focus of extensive research and development, SMS spam detection presents unique challenges. The concise nature of SMS messages, coupled with their limited contextual information, makes it difficult to apply traditional spam detection techniques effectively [7].

Given the financial and social implications of SMS spam, developing robust and efficient detection systems is urgent [8]. These systems must accurately identify and filter spam messages, adapt to evolving spamming techniques, operate efficiently on mobile devices with limited resources, and scale to handle vast amounts of SMS data.

Historically, SMS spam detection began with rule-based systems and blacklists [9]. While easy to implement, these methods quickly became ineffective as spammers adapted their techniques, leading to an arms race between spammers and defenders. Consequently, research shifted to machine learning (ML) algorithms [5], which offered greater flexibility and improved accuracy by learning patterns directly from data. Early ML approaches, such as Naive Bayes, Support Vector Machines (SVMs), and decision trees, were able to capture more complex data patterns [10]. However, these methods

often struggled to capture long-range dependencies and complex word relationships within SMS messages, limiting their effectiveness in accurately distinguishing spam from legitimate messages [11]. As a result, most of these methods have gradually lost their popularity in the SMS spam detection field.

The advent of deep learning (DL) marked a significant advancement, with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) automatically learning hierarchical features from raw text data. These models were capable of capturing more intricate patterns and dependencies within the text, enhancing spam detection performance. Despite their progress, these methods still faced challenges in effectively modeling long-distance dependencies and complex word interactions, which are crucial for understanding the nuanced meanings in SMS messages [12]. More recently, transformers like BERT (Bidirectional Encoder Representations from Transformers) have further enhanced SMS spam detection by providing context-aware embeddings [13]. BERT's self-attention mechanism allows it to capture contextual relationships between words effectively. However, BERT may struggle to effectively model certain local dependencies and complex syntactic structures inherent in some spam messages, potentially limiting its classification accuracy in nuanced scenarios.

Aforesaid research has predominantly focused on leveraging word embeddings to capture semantic and contextual information within text. However, word embeddings like GloVe, fastText, BERT and so on, despite their effectiveness, have inherent limitations in capturing long-range dependencies and complex word relationships. This can hinder the model's ability to fully understand nuanced meanings in SMS messages.

Recent studies have also employed Graph Convolutional Networks (GCNs) for text classification, leveraging global vocabulary information by constructing graphs where words are nodes connected by co-occurrence or other relationships. However, GCNs that only take into account global vocabulary information may fail to capture local information, such as word order, which is crucial for understanding the meaning of a sentence.

To address these limitations, we propose the BERT with Triple-Graph Convolutional Networks (BERT-G3CN) model for SMS spam detection, which integrates BERT-based word embeddings with graph embeddings derived from three distinct graph types: Cooccurrence Graphs, Heterogeneous Graphs, and Integrated Syntactic Graphs. The Cooccurrence Graphs focus on the frequency and patterns of word co-occurrence, enhancing the model's ability to detect recurring spam indicators. The Heterogeneous Graphs capture di-

verse semantic relationships by modeling interactions between different types of entities, such as parts of speech and named entities, thereby enriching the model's semantic understanding. Finally, the Integrated Syntactic Graphs incorporate syntactic dependencies to understand the grammatical structure, ensuring the model grasps the intended meaning and sentiment conveyed through word arrangement. By integrating these three types of graph embeddings, BERT-G3CN effectively captures both global relational patterns and local syntactic information, thereby overcoming the limitations of traditional GCN approaches and enhancing the accuracy and robustness of SMS spam detection.

Our approach provides a comprehensive understanding of the integrated message of a SMS message. For example, consider the SMS message: "Limited time offer available now."

- BERT Word Embeddings: BERT effectively captures the semantic meaning of each word, understanding "limited," "time," "offer," and "available" in context. However, it may not fully grasp "limited time offer" as a cohesive phrase indicating a spammy promotion.
- Cooccurrence Graphs: Identify that "limited," "time," and "offer" frequently co-occur in spam messages, reinforcing their association as indicators of spam.
- Heterogeneous Graphs: Model the relationship between "limited" (adjective) and "offer" (noun), enhancing the understanding that "limited time offer" is a specific promotional tactic.
- Integrated Syntactic Graphs: Capture the syntactic dependency between "limited time offer" and "available now," clarifying that the entire phrase forms a cohesive promotional message aimed at creating urgency.

These graph embeddings are processed through Graph Convolutional Networks (GCNs), which extract meaningful structural features from each graph type. By integrating the global relational information captured by the graphs with BERT's powerful contextual embeddings, our model achieves a comprehensive feature representation. This fusion not only improves the accuracy of SMS spam detection, but also offers a novel perspective on feature extraction, contributing to a deeper understanding and more effective approaches to SMS spam detection.

The main contributions of this paper are as follows:

- First Integration of Triple Graph Embeddings with BERT: we design the BERT-G3CN architecture, the first model to combine BERT-based word embeddings with graph embeddings derived from Cooccurrence, Heterogeneous, and Integrated Syntactic Graphs. This novel integration enables the model to harness the strengths of both contextual and structural information, setting a new benchmark in SMS spam detection.

- Effective Capture of Global and Local Semantic Information: by incorporating three distinct types of graph embeddings, our model adeptly captures both global co-occurrence patterns and local syntactic relationships within SMS messages. The application of Graph Convolutional Networks (GCNs) on each graph type facilitates the extraction of comprehensive structural features that complement BERT's contextual embeddings, thereby enhancing the model's ability to discern nuanced spam indicators.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive review of related works on SMS spam detection, covering the evolution of techniques from Rule-Based and Blacklist Methods to more advanced Machine Learning Approaches, Deep Learning Approaches, Transformer-based Models. In Section 3, we introduce the BERT-G3CN model, detailing its algorithm and design architecture. Section 4 describes the two datasets utilized in our experiments and presents the performance of BERT-G3CN on these datasets, comparing it with other related models. Finally, in Section 5, we conclude this research based on our findings.

2. Related Works

The field of SMS spam detection has seen considerable advancements over the years, evolving from simple rule-based systems to sophisticated machine learning and deep learning approaches. This section reviews the progression of SMS spam detection techniques and explains the novelty of our method accordingly.

2.1. Rule-Based and Blacklist Methods

Early SMS spam detection systems relied heavily on rule-based approaches and blacklist mechanisms. These methods used predefined rules and maintained lists of known spam sources to filter out unwanted messages [14]. While easy to implement, these systems quickly became ineffective as spammers adapted their techniques to bypass the rules and blacklists. Consequently, the accuracy of such systems was limited, leading to a high rate of false positives and negatives.

2.2. Machine Learning Approaches

As the limitations of rule-based systems became apparent, researchers began to explore machine learning (ML) algorithms for SMS spam detection. One of the earliest and most widely adopted ML techniques in this domain is the Support Vector Machine (SVM) algorithm. SVM's capability to find the optimal hyperplane that separates spam from legitimate messages has made it a popular choice for SMS spam detection. For instance, Sjarif et al. [15] successfully implemented SVM in conjunction with the K-Nearest Neighbors (KNN) algorithm, achieving a commendable accuracy of 98.9% for SMS spam detection.

Further advancements were demonstrated by Himani Jain et al. [16], who conducted an extensive analysis of SMS spam messages using various machine learning models. Their study highlighted the effectiveness of Naive Bayes and SVM algorithms, with accuracy rates reaching as high as 99.7% and 97%, respectively. These results underscore the robustness of SVM and Naive Bayes in handling the unique challenges posed by SMS spam detection.

Moreover, Ram Bheemesh K et al. [17] provided a comparative analysis between SVM and Linear Regression. Their findings revealed that SVM, with its superior ability to capture the complex patterns within SMS content, significantly outperformed Linear Regression, leading to higher accuracy in SMS spam detection. However, while these studies report high accuracy rates, they often do not consider the practical challenges of feature engineering and the need for domain expertise, which can limit the generalizability and scalability of these models.

In a comprehensive study, Phani Teja Nallamothu et al. [18] compared various ML models, including Naive Bayes, SVM, Decision Trees, Random Forest, Logistic Regression, and K-Nearest Neighbors, for their effectiveness in SMS spam detection. Their results indicated that SVM and Naive Bayes, both supervised learning algorithms, consistently outperformed other models

in accurately identifying spam. However, this study, like others, still relied on manually crafted features, which may not always capture the full complexity of SMS spam, and require significant domain expertise for optimal performance [11].

2.3. Deep Learning Approaches

The advent of deep learning brought a significant shift in SMS spam detection methodologies. Deep neural networks, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), demonstrated superior performance by automatically learning hierarchical features from raw text data [19]. Kim and Yoon [20] applied CNNs for sentence classification, effectively capturing local patterns in text, which is crucial for identifying specific phrases and words associated with spam.

In addition to CNNs, Long Short-Term Memory (LSTM) networks, a variant of RNNs, have proven particularly effective in capturing sequential dependencies in text data. Gauri Jain, Manisha Sharma et al. [21] pioneered the use of WordNet and ConceptNet word embedding techniques in conjunction with a CNN and LSTM architecture for SMS spam detection in social media. Their approach demonstrated the power of combining advanced word embeddings with deep learning architectures, achieving notable improvements in detection performance. Similarly, Chilukuri Lekhya Sri et al. [22] enhanced SMS spam detection by combining LSTM with a fast encoder, achieving an accuracy of 96.19%. Moreover, Vangapandu Venkata Kalyani et al. [23] proposed an LSTM variant that integrates both LSTM and GRU architectures. Their model demonstrated superior performance compared to other deep learning models, such as Random Forests, RNNs, and standalone LSTMs, achieving an accuracy of 99%.

Recently, transformer-based models have further advanced deep learning's impact on SMS spam detection. Transformer architectures, such as Bidirectional Encoder Representations from Transformers (BERT), have gained attention due to their ability to capture long-range dependencies and provide context-aware embeddings. Devlin et al. [24] demonstrated that BERT's ability to consider the context of each word in a sentence allows it to capture nuanced meanings and relationships, leading to highly accurate SMS spam detection. Fahfouh et al. [25] combined BERT embeddings with CNNs and Gated Recurrent Neural Networks (GRNN) with attention mechanisms to enhance the detection of deceptive opinion spam, further improving the robustness of spam detection systems.

However, despite their effectiveness, existing deep learning and transformer-based models often encounter challenges in capturing both global co-occurrence patterns and intricate semantic relationships within SMS data. Traditional CNNs and RNNs primarily focus on local patterns or sequential dependencies, which may overlook broader contextual information crucial for distinguishing spam messages. Moreover, while transformer-based models like BERT excel at contextual embeddings, they do not explicitly model syntactic dependencies and co-occurrence relationships, potentially limiting their performance in scenarios where such structural information is pivotal.

2.4. Graph Convolutional Networks (GCN) Approaches

Research on applying Graph Convolutional Networks (GCNs) to SMS spam detection is relatively limited. However, inspiration can be drawn from analogous domains where GCNs have been successfully employed to identify spammers and detect spam content.

Zhiwei Guo et al. [26] proposed the Deep Graph Neural Network-based Spammer Detection (DeG-Spam) model, which establishes a social graph neural network framework to process both occasional relations and inherent relations among users. By modeling the intricate social interactions and inherent user characteristics within the social graph, DeG-Spam effectively distinguishes spammers from legitimate users, achieving high detection performance.

Similarly, P. Jayashree et al. [27] introduced the Metapath-based Graph Convolution Network (M-GCN) framework for spam review detection. M-GCN leverages metapaths in heterogeneous networks to capture the complex semantic meanings of reviews, enabling the model to discern subtle patterns indicative of spam. Their approach attained an accuracy of 96% on benchmark datasets from Yelp and Amazon, underscoring the potential of metapath-based GCNs in handling heterogeneous data for spam detection tasks.

Despite these advancements, these models exhibit certain limitations when applied to SMS spam detection. These approaches primarily focus on social graph structures, which may not fully encapsulate the diverse and nuanced textual relationships present in SMS messages. Furthermore, these approaches often rely on predefined metapaths, which may restrict their flexibility and adaptability to evolving spam patterns in SMS data.

Addressing these challenges, the proposed BERT-G3CN model integrates graph-based embeddings that capture global co-occurrence patterns, diverse

semantic relationships, and syntactic dependencies alongside BERT's contextual word embeddings. This multi-faceted approach ensures that both global and local textual patterns are effectively captured, enhancing the model's ability to accurately classify SMS messages as spam or legitimate. By leveraging Graph Convolutional Networks (GCNs) to process distinct graph structures, BERT-G3CN provides a robust and comprehensive framework.

3. Methodology

3.1. Overview

The proposed BERT-G3CN (BERT with Triple-Graph Convolutional Networks) model for SMS spam detection integrates BERT-based word embeddings with graph embeddings derived from three distinct graph types: Cooccurrence Graphs, Heterogeneous Graphs, and Integrated Syntactic Graphs. This multi-graph approach captures diverse and complementary features, enabling the model to leverage both global and local textual patterns for enhanced spam detection. As shown in Figure 1, the model first generates word embeddings from the input text using BERT, followed by constructing and processing the three graph types to capture cooccurrence relationships, semantic diversity, and syntactic dependencies. These graph embeddings are then fused with the word embeddings to form enriched representations, which are refined through self-attention layers before final classification, providing a holistic understanding of SMS content.

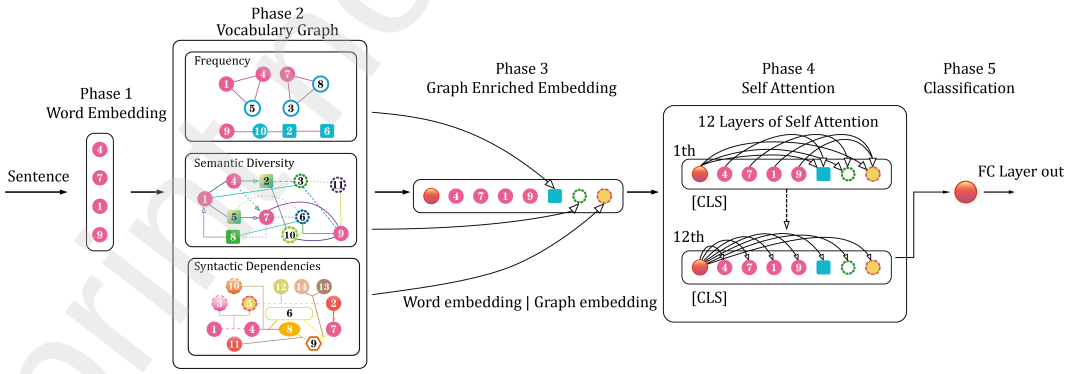

Figure 1: Workflow of BERT-G3CN for SMS Spam Detection.

3.2. Graph Construction

3.2.1. Cooccurrence Graphs

Cooccurrence Graphs are constructed to capture the frequency and patterns of word co-occurrence within SMS messages. In this graph, each unique word in the vocabulary is represented as a node. Edges between nodes indicate that the connected words co-occur within a predefined window size w in the text. The weight of an edge between two words u and v is determined by the frequency of their co-occurrence.

Mathematically, the cooccurrence graph $G_{\text{co}} = (V, E_{\text{co}})$ is defined as:

$$\text{weight}(u, v) = \sum_{i=1}^{N} \sum_{j=i+1}^{i+w} \mathbb{I}(w_i = u \text{ and } w_j = v) \tag{1}$$

where N is the total number of words in the SMS dataset, w_i and w_j are words at positions i and j respectively, and $\mathbb{I}$ is the indicator function that returns 1 if the condition is true, and 0 otherwise.

This graph effectively highlights global relational patterns by identifying frequently co-occurring word pairs, which can serve as strong indicators of spam.

3.2.2. Heterogeneous Graphs

Heterogeneous Graphs aim to model diverse semantic relationships by incorporating multiple types of nodes and edges. Unlike homogeneous graphs, which consist of a single type of node and edge, heterogeneous graphs include various entity types, such as words, parts of speech (POS), and named entities (NER). This enriched structure allows the model to capture interactions between different linguistic elements, providing a deeper semantic understanding of the text.

Formally, the heterogeneous graph $G_{\text{het}} = (V, E_{\text{het}}, R)$ is defined as:

$$G_{\text{het}} = (V, E_{\text{het}}, R) \tag{2}$$

where V includes word nodes, POS nodes, and NER nodes, E_{het} represents edges connecting nodes of different types based on specific relationships, and R denotes the set of relation types, such as word-POS, word-NER, and semantic similarities.

Each relation type $r \in R$ defines a different set of edges E_r:

$$E_r = \{(u, v) \mid \text{relation } r \text{ holds between } u \text{ and } v\} \tag{3}$$

This graph structure captures multifaceted semantic relationships, enabling the model to differentiate subtle nuances between spam and legitimate messages by understanding the roles and interactions of various linguistic entities.

3.2.3. Integrated Syntactic Graphs

Integrated Syntactic Graphs leverage syntactic dependencies to capture the grammatical structure of SMS messages. Utilizing dependency parsing, this graph models the syntactic relationships between words, such as subject-verb and adjective-noun dependencies. By representing these dependencies as edges, the graph ensures that grammatical nuances influencing the message's meaning and sentiment are effectively captured.

Mathematically, the syntactic graph $G_{\text{syn}} = (V, E_{\text{syn}})$ is defined as:

$$G_{\text{syn}} = (V, E_{\text{syn}}) \tag{4}$$

where V represents individual words in the sentence and E_{syn} represents syntactic dependencies between words. Each edge $(u, v) \in E_{\text{syn}}$ denotes a syntactic dependency between words u and v, such as u being the subject of v. This graph captures the grammatical structure, ensuring that the arrangement of words is understood in context, which is crucial for accurately interpreting the intent and sentiment of the message.

3.3. Graph Embedding through Graph Convolutional Networks (GCNs)

Once the three graph types are constructed, Graph Convolutional Networks (GCNs) are employed to extract meaningful structural features from each graph. GCNs operate by aggregating information from a node's neighbors and transforming it through learnable parameters, enabling the model to capture both local and global graph structures.

3.3.1. GCN Architecture

A standard GCN layer is defined by the following equation:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)}\right) \tag{5}$$

where $H^{(l)}$ is the matrix of node features at layer l, $\hat{A} = A + I$ is the adjacency matrix A with added self-connections, $\hat{D}$ is the degree matrix of $\hat{A}$, $W^{(l)}$ is the learnable weight matrix at layer l, and σ is a non-linear activation function (e.g., ReLU).

This formulation ensures that each node's representation is influenced by its immediate neighbors, allowing the GCN to capture localized patterns within the graph.

3.3.2. Processing Each Graph Type

For each graph type, a dedicated GCN is applied to extract relevant features.

Cooccurrence Graphs. For Cooccurrence Graphs G_{co}, the GCN captures global co-occurrence patterns of words. The feature update for layer $l+1$ is given by:

$$H_{\text{co}}^{(l+1)} = \sigma\left(\hat{D}_{\text{co}}^{-1/2}\hat{A}_{\text{co}}\hat{D}_{\text{co}}^{-1/2}H_{\text{co}}^{(l)}W_{\text{co}}^{(l)}\right) \tag{6}$$

where $\hat{A}_{\text{co}}$ and $\hat{D}_{\text{co}}$ are the adjacency and degree matrices for the cooccurrence graph.

Heterogeneous Graphs. For Heterogeneous Graphs G_{het}, a Relational GCN (R-GCN) is employed to handle multiple types of edges. The feature update for layer $l+1$ is defined as:

$$H_{\text{het}}^{(l+1)} = \sigma\left(\sum_{r\in R}\hat{D}_r^{-1/2}\hat{A}_r\hat{D}_r^{-1/2}H_{\text{het}}^{(l)}W_r^{(l)}\right) \tag{7}$$

where R represents the different relation types (e.g., word-POS, word-NER), $\hat{A}_r$ and $\hat{D}_r$ are the adjacency and degree matrices for relation r, and $W_r^{(l)}$ is the weight matrix for relation r at layer l.

This approach allows the GCN to learn distinct representations for each relation type, effectively capturing the diverse semantic relationships within the heterogeneous graph.

Integrated Syntactic Graphs. For Integrated Syntactic Graphs G_{syn}, the GCN focuses on syntactic dependencies. The feature update for layer $l+1$ is given by:

$$H_{\text{syn}}^{(l+1)} = \sigma\left(\hat{D}_{\text{syn}}^{-1/2}\hat{A}_{\text{syn}}\hat{D}_{\text{syn}}^{-1/2}H_{\text{syn}}^{(l)}W_{\text{syn}}^{(l)}\right) \tag{8}$$

where $\hat{A}_{\text{syn}}$ and $\hat{D}_{\text{syn}}$ are the adjacency and degree matrices for the syntactic graph.

This ensures that the GCN captures the grammatical structure and syntactic nuances of the SMS messages, enhancing the model's ability to interpret the intended meaning and sentiment.

3.4. Integration with BERT Word Embeddings

After extracting graph embeddings from each of the three graph types using their respective GCNs, these embeddings are integrated with BERT-based word embeddings to form a comprehensive feature representation for each SMS message.

3.4.1. BERT Word Embeddings

A pre-trained BERT model is utilized to generate contextual word embeddings H_{BERT}:

$$H_{\text{BERT}} = \text{BERT}(X) \tag{9}$$

where X represents the input SMS message tokens. BERT's self-attention mechanism captures the contextual relationships between words, providing rich semantic representations.

3.4.2. Feature Fusion

The graph embeddings H_{co}, H_{het}, and H_{syn} obtained from the Cooccurrence, Heterogeneous, and Integrated Syntactic Graphs are concatenated with the BERT embeddings H_{BERT}:

$$H_{\text{fused}} = \text{Concat}(H_{\text{BERT}}, H_{\text{co}}, H_{\text{het}}, H_{\text{syn}}) \tag{10}$$

To manage the increased dimensionality and ensure consistency, a linear projection is applied:

$$H_{\text{proj}} = \text{ReLU}(H_{\text{fused}} W_{\text{fuse}} + b_{\text{fuse}}) \tag{11}$$

where W_{fuse} and b_{fuse} are learnable parameters. This projection reduces the dimensionality of the fused embeddings and introduces non-linearity, enhancing the feature representation.

3.4.3. Classification Layer

The projected embeddings H_{proj} are then passed through a fully connected layer followed by a softmax activation function to perform the final classification:

$$\hat{y} = \text{Softmax}(H_{\text{proj}} W_{\text{cls}} + b_{\text{cls}}) \tag{12}$$

where W_{cls} and b_{cls} are learnable parameters of the classification layer. The output $\hat{y}$ represents the probability distribution over the classes (spam or legitimate).

4. Experimental results and discussions

In this section, we first introduce the dataset and parameter settings used in our experiments. Next, we detail the configuration of our BERT-G3CN model and the experimental setup. Following this, we compare the performance of our proposed BERT-G3CN model against several baseline methods through comparative experiments. We then present a comparison of different word embedding techniques to evaluate their effectiveness when integrated with the BERT-G3CN model. Finally, we conduct ablation studies to verify the effectiveness of each module within the BERT-G3CN model.

4.1. Dataset

The datasets used for our SMS spam detection experiments are referred to as dataset 1 and dataset 2. Dataset 1 gathered from the UCI repository [28], consists of 5,572 SMS messages: 747 spam and 4,825 ham. Dataset 2, the ExAIS_SMS dataset [29], contains 4,981 SMS messages: 2,740 spam and 2,241 ham. These datasets provide a realistic evaluation of SMS spam detection systems, including a variety of message types and content.

Table 1 presents the distribution of both datasets, highlighting that dataset 1 is heavily imbalanced, with approximately 87% ham messages and 13% spam messages. In contrast, dataset 2 is nearly balanced, with roughly 55% spam and 45% ham messages.

Dataset 2 allows us to assess model performance on a more balanced dataset, contrasting with the imbalanced nature of dataset 1. This comparison provides a comprehensive evaluation of the model's robustness across different class distributions.

Table 1: Comparison of dataset 1 and dataset 2 statistics

Dataset	Label	Number of Instances	Class Distribution (%)
dataset 1	Ham	4825	86.59
	Spam	747	13.41
	Total	5572	100
dataset 2	Ham	2241	44.99
	Spam	2740	55.01
	Total	4981	100

4.2. Configuration

The implementation of the proposed approach was carried out using Python 3.10, utilizing PyTorch. The BERT-G3CN model was built upon the pre-trained BERT model for generating contextual word embeddings. The model integrates three distinct Graph Convolutional Networks (GCNs) to process Cooccurrence Graphs, Heterogeneous Graphs, and Integrated Syntactic Graphs. Each GCN was configured with a hidden dimension of 128 and an output dimension of 64, employing ReLU activation functions and a dropout rate of 0.1 to prevent overfitting.

The GCN embeddings were concatenated with the BERT embeddings and projected back to the original embedding dimension using a linear layer followed by a ReLU activation and a dropout layer with a rate of 0.5. The Transformer encoder consisted of 6 layers with 8 attention heads each, and an attention dropout rate of 0.1.

For the classification layer, a pre-classifier linear layer was used to transform the pooled output, followed by a ReLU activation and a dropout layer with a rate of 0.5. The final classification was performed using a linear layer that maps to the number of target classes (spam or legitimate).

The model was trained for 10 epochs using a batch size of 32 and a learning rate of 2×10^{-5}. The optimizer employed was AdamW with weight decay set to 0.01. The dataset was split with 80% allocated for training and 20% for testing. All experiments were conducted in the Google Colab environment using NVIDIA Tesla V100 GPUs.

Table 2 summarizes the configuration setup of BERT-G3CN.

Table 2: Configuration setup of BERT-G3CN

Parameter	**Value**
BERT Embedding Dimension	768
VGCN Hidden Dimension	128
VGCN Output Dimension	64
Projection Dropout Rate	0.5
Learning Rate	2×10^{-5}
Batch Size	32
Number of Epochs	10

4.3. Evaluation Measures

To assess the effectiveness of our proposed BERT-G3CN model, we utilized several standard evaluation metrics: Accuracy, Precision, Recall, F1-Score, and AUC. These metrics provide a comprehensive view of the model's performance in classifying text.

- **Accuracy** compares the predicted labels with the actual labels:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

  where TP (True Positives) are the number of positive messages correctly identified as positive, TN (True Negatives) are the number of negative messages correctly identified as negative, FP (False Positives) are the number of negative messages incorrectly labeled as positive, and FN (False Negatives) are the number of positive messages incorrectly labeled as negative.

- **Precision** evaluates the ability to avoid labeling negative messages as positive:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

- **Recall** measures the ability to identify all positive messages:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

- **F1-Score** is the harmonic mean of precision and recall:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

- **AUC** (Area Under the Curve) represents the model's ability to distinguish between classes, calculated from the ROC (Receiver Operating Characteristic) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR):

$$\text{AUC} = \text{Area Under the ROC Curve} \tag{17}$$

The use of these metrics ensures a detailed understanding of the classification capabilities of the BERT-G3CN model.

We utilized the sparse categorical cross-entropy as the loss function in our model because it is well-suited for handling classification tasks with multiple classes, especially when dealing with sparse data [30]. This loss function is well-suited for classification tasks where the output is a probability distribution over multiple classes. The sparse categorical cross-entropy is defined as follows:

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^{N} \log p_i \tag{18}$$

where N is the number of samples and p_i is the predicted probability for the true class of the i-th sample.

The performance of BERT-G3CN, as measured by these metrics, is summarized in Table 3. The BERT-G3CN model demonstrates exceptional accuracy and robust performance across all metrics, setting a new standard in SMS spam detection tasks. This comprehensive evaluation experiment shows that BERT-G3CN not only achieves high accuracy but also excels in identifying both spam and ham messages, as reflected in its precision, recall, F1-score, and AUC values.

4.4. Comparative Study

To evaluate the performance between BERT-G3CN and various baselines, we conduct a comprehensive comparative experiment involving traditional machine learning (ML), deep learning (DL), Ensemble, and Transformer-based models. The evaluation metrics include Precision, Recall, F1-score,

Table 3: Performance metrics of BERT-G3CN on dataset 1 and dataset 2

Dataset	Label	Precision	Recall	F1-score	Accuracy
dataset 1	Ham	0.9928	0.9990	0.9959	99.28%
	Spam	0.9925	0.9500	0.9708	
dataset 2	Ham	0.9487	0.9384	0.9435	93.78%
	Spam	0.9244	0.9369	0.9306	

Accuracy. By examining both AUC (Area Under the Curve) and runtime, we achieve a balanced view of the models' effectiveness and feasibility for real-time applications.

We compared several classic ML and DL methods. The traditional ML models include Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbors (KNN), and Gradient Boosting (GB). These models are widely used in SMS spam detection for their robustness. The DL models considered include Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Recurrent Neural Networks (RNN), which are known for their ability to capture complex patterns in data, leading to remarkable performance improvements in various NLP tasks. Then we include the Deep Convolutional Forest (DCF) [31], a model that combines the strengths of deep learning and ensemble methods. DCF utilizes Convolutional Neural Networks (CNN) for feature extraction and employs ensemble methods, specifically random forests, as classifiers to achieve more robust and accurate predictions.

To further highlight the differences between our model and Transformer-based models, we include the BERT model in this experiment. BERT leverages its powerful contextual embeddings combined with a classification head, specifically utilizing a linear layer for classification tasks. This approach enhances the semantic understanding of text while improving the detection accuracy of deceptive opinions.

Table 4 presents a comparative analysis of various models' performance across both dataset 1 and dataset 2, focusing on metrics such as precision, recall, F1-score, and accuracy. On the imbalanced dataset 1, the BERT-G3CN model outperforms all other models, achieving the highest accuracy of 99.28% and a remarkable spam precision of 99.25%. These results demonstrate BERT-G3CN's ability to capture complex patterns in imbalanced data, leading to superior precision and overall accuracy.

Table 4: Comparison of performance metrics for different models on dataset 1 and dataset 2

Dataset	Model	Label	Precision	Recall	F1-score	Accuracy
dataset 1	**RF**	Ham	0.9750	0.9229	0.9483	91.19%
		Spam	0.6094	0.8357	0.7048	
	SVM	Ham	0.9322	0.8900	0.9106	84.73%
		Spam	0.4185	0.5500	0.4753	
	DT	Ham	0.9475	0.8911	0.9184	86.16%
		Spam	0.4646	0.6571	0.5444	
	KNN	Ham	0.9463	0.8520	0.8967	82.84%
		Spam	0.3924	0.6643	0.4934	
	GB	Ham	0.9764	0.8911	0.9318	88.59%
		Spam	0.5289	0.8500	0.6521	
	CNN	Ham	0.9927	0.9815	0.9871	97.75%
		Spam	0.8808	0.9500	0.9141	
	LSTM	Ham	0.9847	0.9897	0.9872	97.75%
		Spam	0.9259	0.8929	0.9091	
	RNN	Ham	0.9927	0.9774	0.9850	97.39%
		Spam	0.8581	0.9500	0.9017	
	DCF	Ham	0.9790	0.9979	0.9883	98.02%
		Spam	0.9877	0.8889	0.9357	
	BERT	Ham	0.9937	0.9958	0.9947	99.10%
		Spam	0.9752	0.9632	0.9691	
	BERT-G3CN	Ham	0.9928	**0.9990**	**0.9959**	**99.28%**
		Spam	**0.9925**	0.9500	**0.9708**	
dataset 2	**RF**	Ham	0.8059	0.3393	0.4775	58.03%
		Spam	0.5099	0.8938	0.6493	
	SVM	Ham	0.5896	0.6021	0.5958	53.82%
		Spam	0.4679	0.4550	0.4614	
	DT	Ham	0.6120	0.5435	0.5757	54.72%
		Spam	0.4819	0.5520	0.5145	
	KNN	Ham	0.5950	0.5062	0.5470	52.61%
		Spam	0.4623	0.5520	0.5032	
	GB	Ham	0.6850	0.4867	0.5691	58.33%
		Spam	0.5151	0.7090	0.5967	
	CNN	Ham	0.9363	0.6004	0.7316	75.10%
		Spam	0.6457	0.9469	0.7678	
	LSTM	Ham	0.9398	0.8313	0.8822	87.45%
		Spam	0.8092	0.9307	0.8657	
	RNN	Ham	0.9462	0.7496	0.8365	83.43%
		Spam	0.7436	0.9446	0.8321	
	DCF	Ham	0.9073	0.9301	0.9186	90.76%
		Spam	0.9080	0.8790	0.8933	
	BERT	Ham	0.9217	0.8877	0.9044	89.26%
		Spam	0.8568	0.8991	0.8774	
	BERT-G3CN	Ham	**0.9487**	**0.9384**	**0.9435**	**93.78%**
		Spam	**0.9244**	0.9369	**0.9306**	

When applied to the more balanced dataset 2, traditional machine learning models like Random Forest, SVM, and Decision Tree show a significant drop in performance. For example, the Random Forest model only achieves an accuracy of 58.03% and a spam precision of 50.99%. This decline likely occurs because these models rely heavily on patterns derived from imbalanced data, which are less effective in a balanced dataset like dataset 2.

Despite the challenges presented by a balanced dataset, BERT-G3CN maintains its leading performance on dataset 2, achieving the highest accuracy of 93.78% and a spam precision of 92.44%. These results confirm that BERT-G3CN, which integrates multiple Graph Convolutional Networks (VGCNs) with BERT's self-attention mechanisms, continues to outperform other models even in more balanced scenarios, maintaining its superior ability in SMS spam detection.

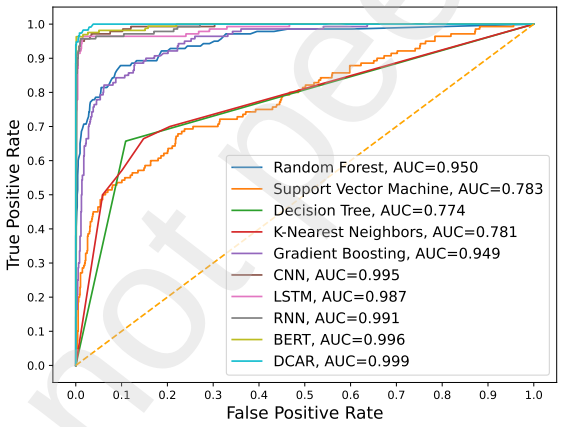

Figure 2: ROC Curves and AUC values for various models on dataset 1

Figures 2 and 3 illustrate the ROC curves and AUC values for various models on dataset 1 and dataset 2. On dataset 1, BERT-G3CN leads with an impressive AUC of 0.999, followed closely by BERT and CNN with AUCs of 0.996 and 0.995, respectively. These results highlight the strength of deep learning and Transformer-based models in handling imbalanced data. In contrast, traditional methods like Random Forest and SVM show weaker performance. On dataset 2, which is more balanced, BERT-G3CN still excels with an AUC of 0.978, while traditional machine learning models such as Random Forest (AUC=0.655) and SVM (AUC=0.541) struggle signifi-

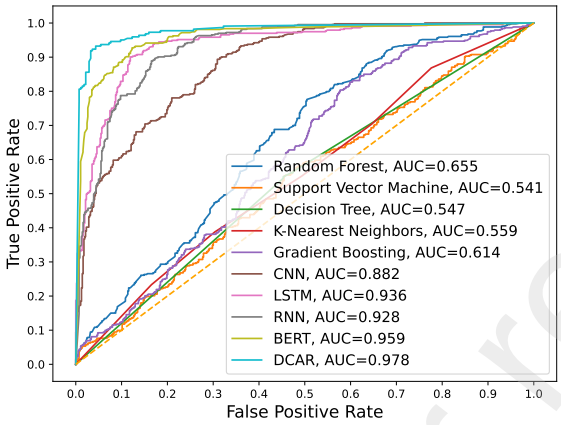

Figure 3: ROC Curves and AUC values for various models on dataset 2

cantly. Overall, BERT-G3CN consistently outperforms other models across both datasets, demonstrating its robustness and effectiveness in SMS spam detection.

The consistency of BERT-G3CN's performance across both datasets underscores its robustness and reliability in SMS spam detection tasks. By integrating multiple Graph Convolutional Networks (VGCNs) with BERT's self-attention mechanisms, BERT-G3CN effectively captures intricate dependencies within the data, allowing it to maintain high levels of precision and recall even in diverse and complex data scenarios.

These results emphasize that BERT-G3CN's integration of multiple Graph Convolutional Networks (VGCNs) with BERT's self-attention mechanisms leads to the highest accuracy, F1-score, and AUC, underscoring its superior performance in SMS spam detection tasks. This comparison highlights the efficacy of BERT-G3CN in capturing complex patterns and dependencies through cooccurrence, heterogeneous, and syntactic graphs, making it a highly effective model for SMS spam detection.

5. Conclusion

This paper presents a BERT with Triple-Graph Convolutional Networks (BERT-G3CN) model for categorizing SMS messages into Spam and Not-Spam. The BERT-G3CN model preprocesses the data by generating con-

textual word embeddings using a pre-trained BERT model and constructs three distinct graph types: Cooccurrence Graphs, Heterogeneous Graphs, and Integrated Syntactic Graphs. Feature extraction is performed using dedicated Graph Convolutional Networks (GCNs) for each graph type, which are then integrated with BERT embeddings to form a comprehensive feature representation. These combined features are subsequently classified using a Transformer-based encoder and a classification head. By leveraging the strengths of both BERT and GCNs, the approach effectively captures complex patterns and dependencies within the data, addressing common challenges faced by existing deep learning methods.

Experimental results demonstrate that BERT-G3CN surpasses traditional machine learning classifiers as well as existing deep learning models and Transformer-based models in terms of precision, recall, F1-score, and accuracy. Our model achieved superior performance compared to existing mainstream SMS spam detection methods on two datasets, with an accuracy of 99.28% and 93.78%, respectively. Overall, the proposed BERT-G3CN model can significantly minimize the risks associated with security threats like SMS phishing by effectively filtering spam messages.

6. Acknowledgments

This work was supported by Key R&D Program of Zhejiang Province (No.2023C01039).

References

[1] D. Miller, L. Abed Rabho, P. Awondo, M. de Vries, M. Duque, P. Garvey, L. Haapio-Kirk, C. Hawkins, A. Otaegui, S. Walton, et al., The global smartphone: Beyond a youth technology, 2021.

[2] S. ul Rehman, R. Gulzar, W. Aslam, Developing the integrated marketing communication (imc) through social media (sm): the modern marketing communication approach, SAGE Open 12 (2022).

[3] A. Nahapetyan, S. Prasad, K. Childs, A. Oest, Y. Ladwig, A. Kapravelos, B. Reaves, On sms phishing tactics and infrastructure, in: 2024 IEEE Symposium on Security and Privacy (SP), 2024, pp. 1–16.

[4] B. Naqvi, K. Perova, A. Farooq, I. Makhdoom, S. Oyedeji, J. Porras, Mitigation strategies against the phishing attacks: A systematic literature review, Computers & Security 132 (2023) 103387.

[5] M. Salman, M. Ikram, M. A. Kaafar, Investigating evasive techniques in sms spam filtering: A comparative analysis of machine learning models, IEEE Access 12 (2024) 24306–24324.

[6] M. Liu, Y. Zhang, B. Liu, Z. Li, H. Duan, D. Sun, Detecting and characterizing sms spearphishing attacks, in: Proceedings of the 37th Annual Computer Security Applications Conference, 2021.

[7] P. Paul, S. Sarkar, G. Manju, Cognitive information-based sms spam detection and filtering of transliterated messages, International Journal of Public Sector Performance Management (2024).

[8] U. Srinivasarao, A. Sharaff, Machine intelligence based hybrid classifier for spam detection and sentiment analysis of sms messages, Multimedia Tools and Applications (2023) 1–31.

[9] A. Mewada, R. K. Dewang, A comprehensive survey of various methods in opinion spam detection, Multimedia Tools and Applications 82 (2022) 13199–13239.

[10] B. N. Sai, B. Swaminathan, Using the k-nearest neighbors algorithm and logistic regression to improve accuracy, a novel machine learning approach for detecting sms spam message, Journal of Survey in Fisheries Sciences 10 (1S) (2023).

[11] A. A. Akinyelu, Advances in spam detection for email spam, web spam, social network spam, and review spam: Ml-based and nature-inspired-based techniques, Journal of Computer Security 29 (5) (2021) 473–529.

[12] S. Hanifi, A. Cammarono, H. Zare-Behtash, Advanced hyperparameter optimization of deep learning models for wind power prediction, Renewable Energy 221 (2024) 119700.

[13] Y. Guo, Z. Mustafaoglu, D. Koundal, Spam detection using bidirectional transformers and machine learning classifier algorithms, journal of Computational and Cognitive Engineering 2 (1) (2023) 5–9.

[14] T. Gangavarapu, C. Jaidhar, B. Chanduka, Applicability of machine learning in spam and phishing email filtering: review and approaches, Artificial Intelligence Review 53 (7) (2020) 5019–5081.

[15] N. N. A. Sjarif, Y. Yahya, S. Chuprat, N. Azmi, Support vector machine algorithm for sms spam classification in the telecommunication industry, Int. J. Adv. Sci. Eng. Inf. Technol 10 (2) (2020) 635–639.

[16] H. Jain, M. Mahadev, An analysis of sms spam detection using machine learning model, in: 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2022, pp. 151–156.

[17] R. B. K, D. N, Accurate sms spam detection using support vector machine in comparison with linear regression, in: 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2023, pp. 1–4.

[18] P. Teja Nallamothu, M. Shais Khan, Machine learning for spam detection, Asian Journal of Advances in Research 6 (1) (2023) 167–179.

[19] S. Kaddoura, S. A. Alex, M. Itani, S. Henno, A. AlNashash, D. J. Hemanth, Arabic spam tweets classification using deep learning, Neural Computing and Applications 35 (23) (2023) 17233–17246.

[20] Y. Kim, Convolutional neural networks for sentence classification, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751.

[21] G. Jain, M. Sharma, B. Agarwal, Spam detection in social media using convolutional and long short term memory neural network, Annals of Mathematics and Artificial Intelligence 85 (1) (2019) 21–44.

[22] C. L. Sri, D. Dhana Lakshmi, K. Ravali, V. Kukreja, S. Hariharan, Improved spam detection through lstm- based approach, in: 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2024, pp. 1–6.

[23] V. V. Kalyani, M. V. Rama Sundari, S. Neelima, P. S. Satya Prasad, P. PattabhıRama Mohan, A. Lakshmanarao, Sms spam detection using nlp and deep learning recurrent neural network variants, in: 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), 2024, pp. 92–96.

[24] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019.

[25] A. Fahfouh, J. Riffi, M. A. Mahraz, A. Yahyaouy, H. Tairi, A contextual relationship model for deceptive opinion spam detection, IEEE Transactions on Neural Networks and Learning Systems 35 (2022) 1228–1239.

[26] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, A. Shalaginov, Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace, Future Generation Computer Systems 117 (2021) 205–218.

[27] P. Jayashree, K. Laila, A. Amuthan, Spam review detection with metapath-aggregated graph convolution network, J. Intell. Fuzzy Syst. 45 (2) (2023) 3005–3023.

[28] T. A. Almeida, et al., Contributions to the study of sms spam filtering: New collection and results, in: Proceedings of the 11th ACM Symposium on Document Engineering, 2011, pp. 259–262.

[29] O. O. Abayomi-Alli, S. Misra, A. Abayomi-Alli, A deep learning method for automatic sms spam classification: Performance of learning algorithms on indigenous dataset, Concurrency and Computation: Practice and Experience 34 (2022).

[30] A. Kumar, G. Vashishtha, C. Gandhi, H. Tang, J. Xiang, Sparse transfer learning for identifying rotor and gear defects in the mechanical machinery, Measurement 179 (2021) 109494.

[31] M. A. Shaaban, Y. F. Hassan, S. K. Guirguis, Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text, Complex & Intelligent Systems 8 (6) (2022) 4897–4909.