Hindawi
Scientific Programming
Volume 2022, Article ID 6737080, 8 pages
https://doi.org/10.1155/2022/6737080

Research Article

Semantic Graph Neural Network: A Conversion from Spam Email Classification to Graph Classification

Weisen Pan,[1] **Jian Li**,[1] **Lisa Gao**,[1] **Liexiang Yue**,[2] **Yan Yang**,[2] **Lingli Deng**,[2] **and Chao Deng**[2]

[1]*China Mobile Technology (USA) Inc., Milpitas, CA, USA*
[2]*China Mobile Research Institute, Beijing, China*

Correspondence should be addressed to Weisen Pan; weisenpan@chinamobile.com

Received 25 October 2021; Revised 26 November 2021; Accepted 3 December 2021; Published 7 January 2022

Academic Editor: Sikandar Ali

In this study, we propose a method named Semantic Graph Neural Network (SGNN) to address the challenging task of email classification. This method converts the email classification problem into a graph classification problem by projecting email into a graph and applying the SGNN model for classification. The email features are generated from the semantic graph; hence, there is no need of embedding the words into a numerical vector representation. The method performance is tested on the different public datasets. Experiments in the public dataset show that the presented method achieves high accuracy in the email classification test against a few public datasets. The performance is better than the state-of-the-art deep learning-based method in terms of spam classification.

1. Introduction

In recent years, unsolicited spam emails have become a big problem over the Internet. Spam emails not only consume a large amount of network bandwidth but also waste users' time dealing with them. Some spam emails may also include malware programs that can gather personal information and relay it to advertisers and other third parties.

Thus, there is a strong need for the development of a more efficient filter to automatically detect such emails. Some researchers have proposed different email spam classification methods. These methods include Naïve Bayes [1], decision tree [2], and support vector machine [3] techniques. These traditional methods usually need to manually extract features as embedding vectors from the emails and feed them into the classification model. Recently, there also have been studies using a convolutional neural network (CNN) [4] for spam email classification [5–8]. CNN models have automatically feature extraction and classification in the whole model, which requires no need of manually extracting features from the emails. Both the traditional and CNN methods use the embedding vectors as input. We propose an algorithm in this study that converts the email classification problem into a graph classification problem. Unlike existing methods, our method does not have the step of embedding the email text into the numerical vector representation. Instead, the method projects the content of the email into a graph and uses the graph neural network (GNN) to classify the spam email. The proposed architecture achieved a higher precision for email classification testing against a few public datasets. To summarize, our contributions in this study are as follows:

(i) We present a novel graph neural network-based method for email classification. Our method converts the email classification problem into a graph node classification problem by projecting the email document into the graph. To build the semantic graph network, we employ LDA to automatically discover topic nodes that are contained within a text document. The semantic graph structure can enable the nodes of a word to learn more accurate representation through different collocation.

(ii) The experimental results on different public datasets demonstrate that our proposed algorithm outperforms state-of-the-art email classification methods.

Our method does not need embedding the email text into numerical vector representation and learning predictive word and automatically text embedding.

We organize this study as follows. Section 2 presents the problem statement. Section 3 describes the related work concerning the rule-based method and deep learning-based method. Section 4 discusses our proposed algorithm to employ the GNN to classify spam emails. Section 5 elaborates on the experiments that consist of preprocessing, training, and application of graph neural network, its testing on datasets, and the performance evaluation. We conclude this study in Section 6.

2. Problem Statement

Email classification is the task of assigning tags (ham or spam) to an email according to its content. In the email classification, we are given a description $e \in E$ of an email, where E is a type of high-dimensional email space and a fixed set of classes $C = \{c_1, c_2, \ldots, c_i\}$. In this task, we only have two classes, namely, ham and spam. We are given a training set T of labeled emails $\langle e, c\rangle$, where $\langle e, c\rangle \in E \times C$. For example,

$$\langle e, c\rangle = \langle \text{Congratulations, claim your free \$100 gift card, spam}\rangle. \quad (1)$$

Using a supervised machine learning algorithm, we wish to learn a classification function δ that maps emails to labels.

$$\sigma: E \longrightarrow C. \quad (2)$$

We denote the supervised machine learning method by $\mathscr{L}$ and write $\mathscr{L}(T) = \sigma$. The supervised machine learning method $\mathscr{L}$ takes the training set T as input and returns the learned classification function σ.

3. Related Works

This section introduces related works about email classification in detail. We summary the related work into two methods: the rule-based method and the deep learning-based method.

3.1. Rule-Based Method. To effectively handle the threat posed by spam emails, many researchers have proposed rule-based email classification techniques based on support vector machine and Naïve Bayes theorem and technology. Rathod and Pattewar presented a Naive Bayes method for email classification. The proposed method uses the tokens with ham and spam to calculate the probability to decide whether a mail is a spam or not [9]. The Naïve Bayes method is mainly famous for open-source spam email filters [10]. It is not susceptible to irrelevant features. The reason is that Naïve Bayes usually needs less speedy assessment and training time to detect and filter a spam email. In order to test the Naïve Bayes method for email spam classification, Fitriah et al. [11] use the WEKA [12] tool based on Spambase and Spam datasets for evaluation of the Naïve Bayes method. The experimental result proved that the dataset's number of instances and email type influenced the Naïve Bayes' performance [11]. Support vector machines as one of the most effective classification methods also have been proved over the years. Feng et al. proposed a Naïve Bayes filtering system based on a support vector machine. When the Naive Bayes method is applied, it aims to eliminate the assumption of independence between the features extracted from the input training set. Experimental results show that this method can achieve faster classification speed and higher spam detection accuracy [13]. Vishagini and Rajan proposed to use a weighted support vector machine to filter spam and use the weight variable got from the KFCM algorithm. The weight variable reflects the importance of different categories. The increase in the weight value can reduce the email misclassification. Experiments show that the performance of the spam detection system still needs to be improved in terms of precision and accuracy [14]. Karhika and Visalakshi described a method of spam classification implementing and combining and implementing the ant colony optimization and support vector machine methods. The proposed method is a hybrid model. The model relies on the features of selecting. The experiment shows that the presented algorithm is superior to some of the most advanced classification methods in terms of precision, accuracy, and recall [15]. The advantage of the support vector machine method lies in its high accuracy. However, this method usually is not as fast as other methods.

3.2. Deep Learning-Based Method. Recently, CNN has proven successful in computer vision applications, such as object detection, face recognition, and image classification. Some researchers have employed CNN to solve the spam detection problem. To classify emails as nonphishing and phishing, Bagui et al. proposed a method that uses deep learning technology to capture the inherent characteristic of email. They use one-hot encoding with and without phrases for deep semantic analysis and use deep learning technology to classify emails. They also compared the accuracy of different deep learning and machine learning methods without and with phrases [16]. By analyzing the entire content (i.e., test and images), Seth et al. use a CNN to process it through an independent classifier, and the mail is classified as spam or ham. Two-hybrid multimodal architectures were proposed by them. The architectures collected the input from those two different models and then combined the output information to identify the spam and ham email. Experiments show that the presented method has high accuracy at the classification task than the separated image and text classifiers [17]. An artificial neural network model for email classification is proposed by Alghoul et al. The model is trained using a feedforward backpropagation algorithm. The factors for this model come from Hewlett Packard Labs, George Forman, and Mark Hopkins. This study shows the potential of artificial neural networks in email classification [6]. Soni presented another spam recognition model called THEMIS. To assess the adequacy of THEMIS, they used an unbalanced dataset with a reasonable proportion of phishing and real emails. The experiments showed a promising outcome from the THEMIS model [7]. Srinivasan et al.

proposed a network threat situational awareness framework called DeepSpamNet, which is a powerful and scalable content-based spam detection architecture. Deep learning allows rapid modification of the diverse nature of spammers due to the lack of feature engineering steps. Experiments show that compared with classic machine learning classifiers, the performance of deep learning models is better [8]. The CNN is advantageous because of its self-learning ability and reliable fault tolerance.

3.3. Graph Neural Network. Graph neural network has recently received growing attention [18]. GNN is a type of machine learning algorithm that can extract important information from graphs and make useful predictions. It receives the formatted graph data as input and produces a vector of numerical values that represent relevant information about nodes and their relations, with graphs becoming more pervasive and richer with information, and artificial neural networks becoming more popular and capable. Recently, GNN has become a powerful tool for many natural language preprocessing tasks such as machine translation, social recommendation, and relation classification. In order to do the rating prediction, Fan et al. proposed a GNN-based model that can differentiate the tie strengths by considering social relation heterogeneous strengths. They provide a principled approach to jointly capture interactions and opinions in the user-item graph. The experiments show that the information of opinion plays a very important role in the model performance improvement [19]. To classify relations from clinical notes, Li et al. employ recurrent neural networks and segment graph convolutional to classify relations from clinical notes. They use the dependency syntax of five segments and word sequence with a sentence to build the Seg-GCRN model to learn the relation representations. The experiments demonstrate that the presented algorithm reaches state-of-the-art results for all three relation categories [20]. Bastings et al. proposed an effective and simple method to integrate syntax into a machine translation model for machine translation. The proposed method uses source sentences that predicted syntactic dependency trees to produce word representations. These representations usually are very sensitive to syntactic neighborhoods. They evaluate the performance with Czech-English and German-English translation experiments. The result shows substantial improvements over the syntax agnostic versions in the considered setups [21]. All those previous works either viewed a sentence or a document as a word node of a graph or relied on the relation of document citation to constructing a graph. When constructing the sematic neural graph in this study, we not only consider the words and email as nodes but also employ LDA to automatically discover topic nodes to enrich the semantic information. The main advantage of graph neural network-based algorithms is that the graph neural networks are able to capture the graph structure of data. In addition, the graph neural network can also capture the rich relation information among elements and provide an easy way to do graph-level, edge-level, and node-level prediction tasks.

4. Proposed Architecture

In this study, we convert the email classification problem into a graph classification problem by projecting email into a graph. The email features are generated from the semantic graph; hence, there is no need of embedding the words into a numerical vector representation. The proposed method converts the spam email classification problem into a graph classification problem. As shown in Figure 1, the proposed solution consists of four major phases, data preprocessing, graph building, graph neural network training testing, and graph classification. The dataset is noisy and unbalanced; hence, the dataset needs to be manually cleaned by using data preprocessing techniques. Then, we build a large graph that consists of email document nodes and word nodes. Each node includes embedding vectors based on the properties of their neighbor nodes. We feed the graph to the GNN to learn high-dimensional features after constructing the graph. Finally, we turn the email classification problem into a graph classification based on the email document and word graph convolutional neural network.

4.1. Data Preprocessing. Data preprocessing is needed for transferring email from human language to machine-readable format for further processing. As shown in Figure 2, we perform a series of steps for data preprocessing that include the following: removing punctuations, converting all letters to lower case, removing stop words, tokenizing, and stemming.

(i) Remove punctuations if they are not relevant to the analysis.

(ii) Convert letters to lower case: it can help to reduce the vocabulary size for the input text data.

(iii) Remove stop word: it is the process of getting rid of common words such as prepositions and pronouns. The reason is those stop words are frequent and widespread, hence not providing much information about the corresponding text.

(iv) Tokenization: it is the process of segmenting the input email text into words and sentences. It is quite simple in English that separate words by a blank space.

(v) Stemming: it is an approach to normalize text data and get words to match each other if they are not in the same tense. The stemming removes affixes at the end and the beginning of the words through string operation.

4.2. Building Graph. To classify the spam email, we build an email text graph that includes email document nodes, word nodes, and topic nodes. The graph is defined as

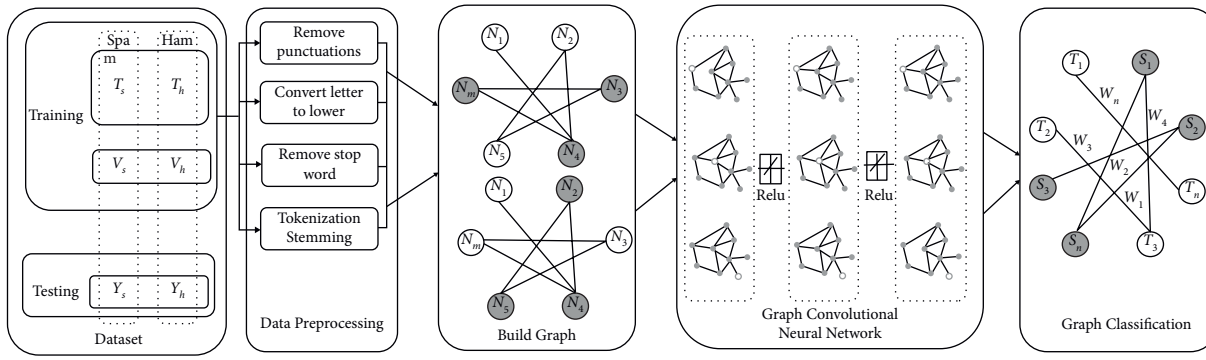

Figure 1: The proposed architecture for spam email classification.

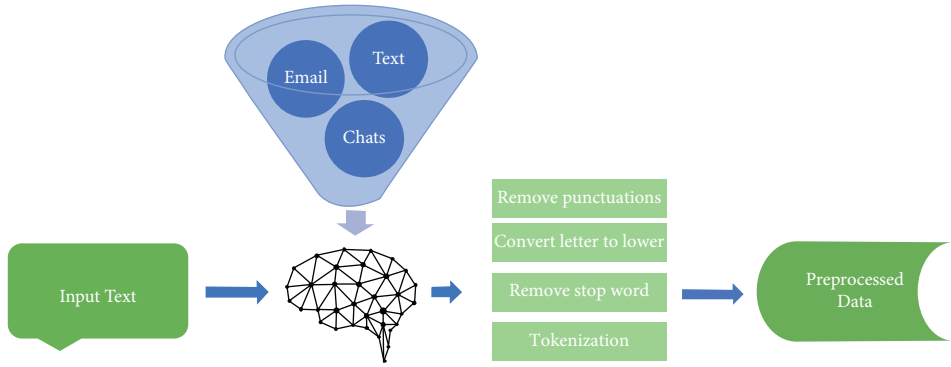

Figure 2: The data preprocessing.

$$\begin{aligned} G &= (V, E), \\ V &= \{\text{word}|\text{text}|\text{topic}\}, \\ E &= \begin{cases} e_{ij} \,|\, e_{id} \,|\, e_{jd}, & i \text{ is word}, \\ j \text{ is text}, & d \text{ is domain topic}, \end{cases} \end{aligned} \tag{3}$$

where V denotes the sets of nodes and E denotes sets of edges. There are three types of nodes: word nodes, email text nodes, and topic nodes. We employ the latent Dirichlet allocation (LDA) [22] model to learn the domain topic from the email documents. LDA is a generative probabilistic model that can cluster the latent semantic structure of the corpus. We use LDA to help us automatically discover topics that are contained within a text document. The difference between the topic node and word node is that the topic node is learned from the email documents using the LDA algorithm. The word node is directly obtained from the email documents.

For each topic d, LDA learns a topic-word joint distribution. Given the parameters α and β, the joint distribution of the topic mixture θ, a set of N words ω, and a set of N topic z are given by

$$\rho\left(\frac{\theta, z, \omega}{\alpha, \beta}\right) = \rho\left(\frac{\theta}{|\alpha}\right) \prod_{n=1}^{N} \rho\left(\frac{z_n}{\theta}\right) \rho\left(\frac{\omega_n|}{Z_n}\right), \beta. \tag{4}$$

The edges in the graph consist of the word-word edges, the word-text edges, the topic-word edges, and the topic-text edges. The weights of the topic-text edges and topic-word edges are gotten by the LDA model. We calculate the word-word edge weights by employing the Pointwise Mutual Information (PMI). The idea of PMI is that we want to quantify the likelihood of co-occurrence of two words. A high PMI score indicates a strong semantic correlation of words, while a low PMI score implies a weak semantic correlation. The formula for PMI is

$$\begin{aligned} \text{PMI}(a, b) &= \log 2^{\rho(a,b)/\rho(a)\rho(b)}, \\ \rho(a) &= \frac{W(a)}{|W|}, \\ \rho(a, b) &= \frac{W(a, b)}{|W|}, \end{aligned} \tag{5}$$

where a and b are a pair of word. $W(a, b)$ is the number of a sliding window containing both the word a and b. $W(a)$ is the number of the sliding window only containing the word a in the corpus. We only keep the edges with the positive PMI values while excluding the edges with negative PMI values.

We employ the BM25 algorithm [23] to calculate the weight of the edge between word and text. BM25 is a bag-of-words search function, which sorts a set of documents according to query terms. Given a query word ω, the BM25 score of the document d is

$$\text{relevant}_{\text{score}(w,\text{doc})} = \sum_{i=1}^{n} \text{IDF}(q_i) \times \frac{TF(q_i) \times (k_1 + 1)}{TF(q_i) + k_1 \times (1 - b + b \times |\text{doc}|/\text{ave\_len})}, \tag{6}$$

where $\text{IDF}(q_i)$ is q_i 's inverse document frequency in the document doc. The inverse document frequency can be obtained by dividing the total number of documents by the number of documents containing the term in the given corpus. It is a numerical statistic if a term is common or rare in the corpus. $TF(q_i)$ is q_i 's term frequency. The term frequency denotes the word number of times that appears in the given document. |doc| is the length of the document doc. ave_len is the average length in the given document. b and k_1 are free parameters.

4.3. Graph Neural Network Mechanism. With the constructed graph representation, we convert the email classification problem into a graph node classification problem by projecting the email document into the graph. Recently, graph neural network is proved to have a convincing performance on such problems [24, 25]. Graph neural network is proposed to collect aggregate information from graph structure. Unlike traditional neural network, GNNs retain a state that can represent information from their neighborhood with arbitrary depth. The purpose of GNN is to learn a state embedding h_v, which is defined as

$$h_v = f\left(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}\right), \tag{7}$$

where h_v contains the information of the neighborhood of each node. It is an n-dimension vector of node v. x_v is the feature of node v. $x_{co[v]}$ is the feature of the edge. $h_{ne[v]}$ is the state information of the node. $x_{ne[v]}$ is the node feature in the neighborhood of v. f is a local transition parametric function. In this study, a three-layer graph neural network is employed for graph classification. The architecture of the three-layer GCN model is expressed as

$$Y = \text{soft}\max\left(\text{H}\sigma\left(H\sigma\left(HXW^{(1)}\right)W^{(2)}\right)W^{(3)}\right), \tag{8}$$

where σ is the activation function $\text{Relu}(x_i) = \max(0, x)$. Y denotes the final result of classifiers. $W^{(1)}$, $W^{(2)}$, and $W^{(3)}$ are weight matrices that are trained using gradient descent. $H = \widetilde{D}^{1/2}\widetilde{A}\widetilde{D}^{1/2}$ denotes Laplacian matrix, $\widetilde{A} = A + I$. A is an adjacency matrix, and I is an identity matrix. $\widetilde{D}$ is the degree matrix of $\widetilde{A}$.

The target of training is to minimize the cross-entropy loss between the predated label and the ground truth label. The loss function is defined as follows:

$$\text{loss} = -\sum_{l \in Y_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}, \tag{9}$$

where Y_L is the label of ground truth; Y is the label indicator matrix; F is the output feature dimension; and Z is the output matrix.

4.4. Node Classification. In this task, there are only two categories, namely, spam and ham. For the new input email, we build the input graph using the Section 4.2 method. And then, we feed the graph to the pretrained model to predict the category. Figure 3 shows the schematic for the node classification. We combine all the embedding (node embedding, edge embedding, and adjacent embedding) to predict the new node. We use the 256-dimensional feature to make the prediction.

5. Experiment

In this section, we introduce our experimental setup and implementation details and conduct several experiments in different public datasets to evaluate our method.

5.1. Experimental Setup. For experiments, we utilize public datasets including Enron-Spam, Spambase, and TREC Spam datasets. The overview of datasets is listed in Table 1.

(i) Enron-Spam dataset: the Enron-Spam dataset was obtained by the Federal Energy Regulatory while investigating the collapse of Enron. It contains approximately 500,000 emails generated by the employees of Enron [26].

(ii) Spambase dataset: this dataset focuses on classifying email as spam or nonspam by frequency of word or character. The dataset contains 4,601 instances and 58 variables. It contains two fields spam and not spam for prediction. It is a multivariate, real dataset mainly used for the classification of attributes. The dataset was developed at Hewlett Packard Labs and was donated by George Forman [27].

(iii) TREC Spam dataset: in this dataset, each email is labeled as nonspam and spam by a chronological index. The dataset contains 92,189 email messages. A total of 39,399 messages are labeled ham, while 52,790 are labeled spam [28].

5.2. Implementation Details. We conduct several experiments to evaluate the performance of the proposed model. We set the node representation dimension as 128 and initialize with Glove [29]. We also vary feature dimensions in further experiments. We set the L2 weight decay to 10^{-5} and set the learning to 10^{-3}. The training batch size is set to 64 and uses the Adam optimizer to train the model, which is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. The Adam optimizer combines the best properties of the root mean square propagation and adaptive gradient algorithms to provide an

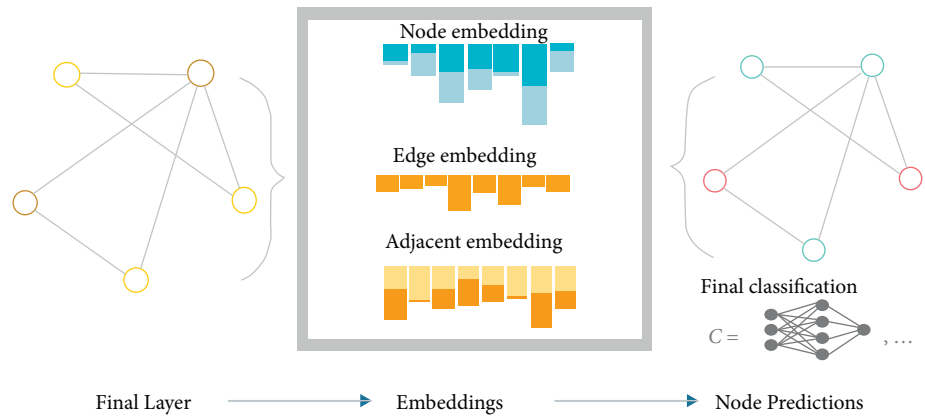

Figure 3: Schematic for node classification.

Table 1: Dataset overview.

Datasets	Number of train	Number of test	Number of validation
Enron-Spam	350,000	50,000	100,000
Spambase	3,220	920	460
TREC	64,532	9218	12,906

Table 2: Results from our SGNN method against CNN-based model on different datasets.

Model	Enron-Spam (%)	Spambase (%)	TREC Spam (%)
CNN-based [8]	93.718	94.193	93.241
Our SGNN	97.872	98.014	96.573

optimization algorithm that can handle sparse gradients on noisy problems and works well on deep neural networks.

5.3. Experimental Results. We compare our algorithm with the most advanced deep learning-based email classification model, and the results are reported in Table 2. It is clear that our proposed model consistently outperforms the state-of-the-art model by more than 3 percentage points (see Figure 4). The result of the graph-based model is better than traditional models like CNN. The main reason why SGNN works well is that SGNN can not only capture word-word relations but also capture the word-topic relations. The other reason is due to the characteristics of the graph structure. It allows a different number of neighbor nodes to exist that can enable the nodes of the word to learn more accurate representation through different collocations.

The other advantage of our method is that our simple projection of text to graph is easy to implement and very robust. The users do not even need to perform complicated data preprocessing. The experimental results of the SGNN demonstrate that the effect of classification of email can be improved by using word-topic semantic information.

The accuracy of our proposed model at different feature dimensions, on different public datasets, is presented. Figures 5–7 show the test accuracy for the Enron-Spam, Spambase, and TREC Spam datasets, respectively. We vary feature dimensions from 32 to 1,024 and report the results of email classification tasks on three datasets. The testing accuracy is improved on all three datasets when the feature dimension increases. The results show that SGNN is stable after the feature dimension is greater than 256. It can also be observed that different feature numbers on different datasets

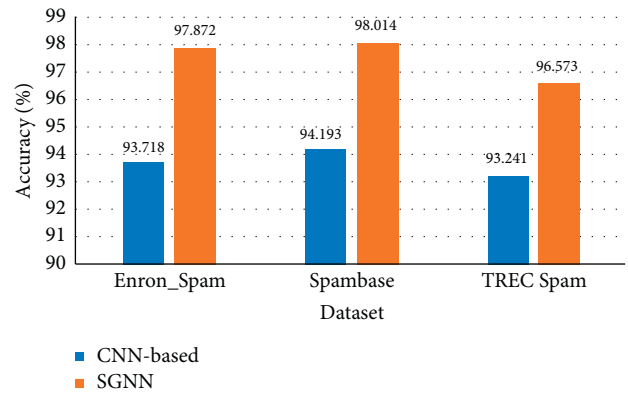

Figure 4: Accuracy of different datasets.

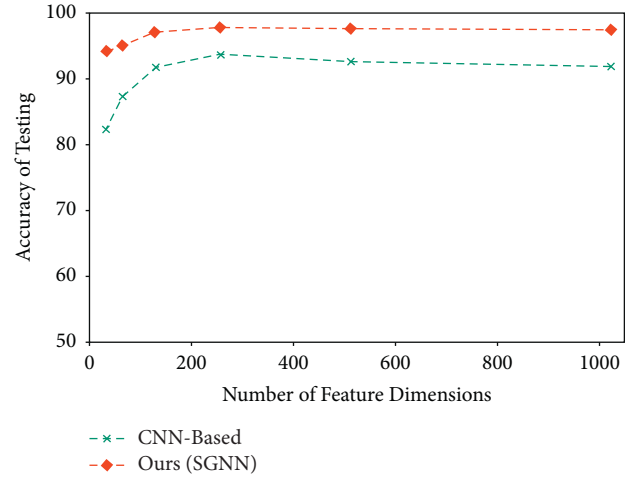

Figure 5: Accuracy of testing on Enron-Spam dataset.

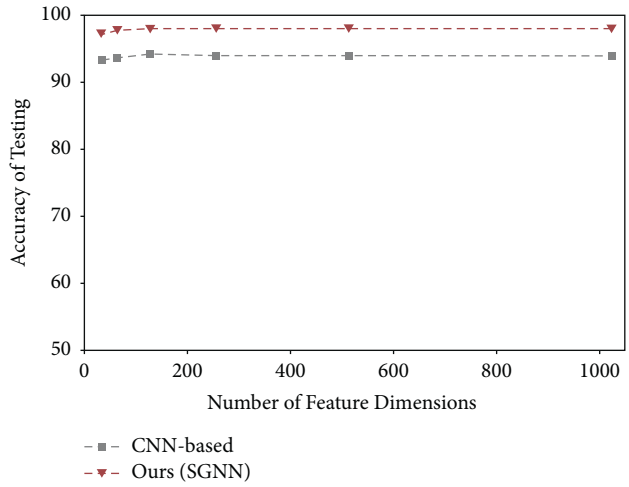

Figure 6: Accuracy of testing on Spambase dataset.

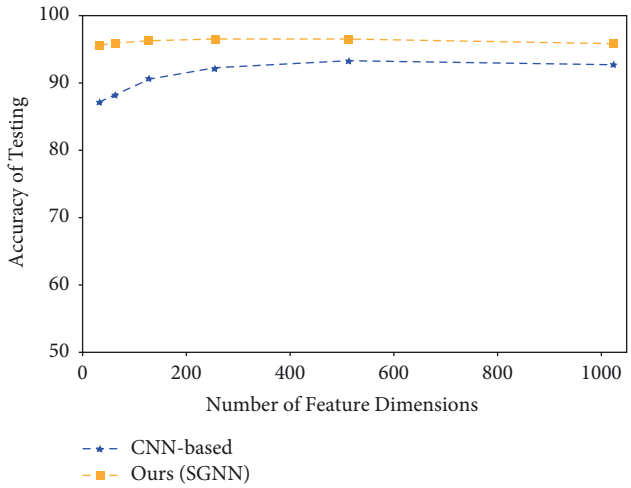

Figure 7: Accuracy of testing on TREC Spam dataset.

led to different classification effects. The main limitation of the proposed method is not robust to noise in graph data. Adding a slight noise in the graph through edge addition or node perturbation is having an adversarial effect on the output of the proposed semantic graph natural network.

5.4. Analysis of Training Time and Memory Consumption. In this section, we report results for the training time per epoch including the forward and backward for 100 epochs on the graphs and measured in seconds in wall-clock time. The above section describes the detailed description of the public dataset used in this experiment. In this experiment, we compare the result of a CPU-only and a GPU implementation. It can be observed from Figure 8 that as the edges increase, the GPU has a faster training speed.

We compare the memory consumption between our model and the CNN-based model shown in Table 3. From the table, we can see that our model has a significant advantage in memory consumption.

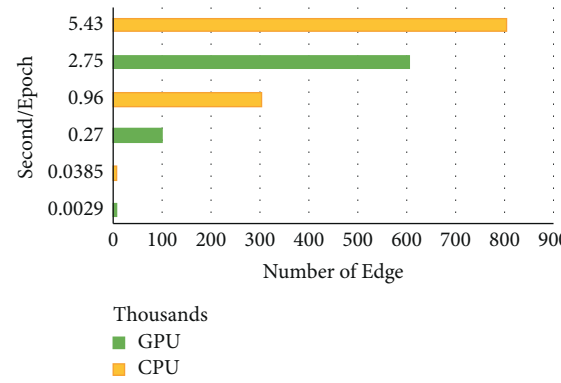

Figure 8: Wall-clock time per epoch for graph.

Table 3: Memory consumption comparison.

Datasets	CNN-based (M)	Our model (M)
Enron-Spam	14,280	6892
Spambase	1681	925
TREC	7893	4287

6. Conclusions and Future Works

In this study, we propose an SGNN method for email classification. It converts the email classification problem into graph classification and then applies the GNN model to classify the email. The features of the email are aromatically extracted by the GNN model. We have tested our method on different public datasets. The experimental results showed that our performance is better than the state-of-the-art deep learning-based method in terms of spam classification. For future work, we can apply various preprocessing techniques such as word disambiguation and other methods to further increase the accuracy of the proposed method. Currently, the proposed method is only applicable to text-based email spam detection. We plan to extend our SGNN approach and make it suitable for filtering spams with different types of data in the future.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interests regarding the publication of this study.

References

[1] A. Harisinghaney, A. Dixit, S. Gupta, and A. Arora, "Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm," in *Proceedings of the 2014 International Conference on Reliability Optimization and Information Technology*, pp. 153–155, IEEE, Faridabad, India, February 2014.

[2] A. Sharaff and H. Gupta, "Extra-tree classifier with metaheuristics approach for email classification," *Advances in Computer Communication and Computational Sciences.Advances in Intelligent Systems and Computing*, Springer, vol. 924, pp. 189–197, Singapore, 2019.

[3] N. Bouguila and O. Amayri, "A discrete mixture-based kernel for SVMs: application to spam and image categorization," *Information Processing & Management*, vol. 45, no. 6, pp. 631–642, 2009.

[4] A. Derhab, A. Aldweesh, A. Z. Emam, and F A. Khan, "Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6689134, 16 pages, 2020.

[5] Y. Cao, X. Liao, and Y. Li, "An e-mail filtering approach using neural network," in *Proceedings of the International Symposium on Neural Networks.Lecture Notes in Computer Science*, pp. 688–694, Moscow, Russia, July 2004.

[6] A. Alghoul, A. Sara, J. Ghada, H. Ghayda, and A. N. Samy, "Email classification using artificial neural network," *International Journal of Applied Engineering Research*, vol. 2, no. 11, 2018.

[7] A. N. Soni, "Spam e-mail detection using advanced deep convolution neural network algorithms," *Journal for Innovative Development in Pharmaceutical and Technical Science*, vol. 2, no. 5, pp. 74–80, 2019.

[8] S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A. M. Al-Zoubi, and S. Kotti Padannayil, "Spam emails detection based on distributed word embedding with deep learning," *Machine Intelligence and Big Data Analytics for Cybersecurity Applications.Studies in Computational Intelligence*, Springer, Cham, New York, NY, USA, pp. 161–189, 2021.

[9] S. B. Rathod and T. M. Pattewar, "Content based spam detection in email using Bayesian classifier," in *Proceedings of the 2015 International Conference on Communications and Signal Processing*, pp. 1257–1261, IEEE, Chengdu, China, April 2015.

[10] I. Androutsopoulos, G. Paliouras, and E. Michelakis, "Learning to filter unsolicited commercial e-mail," *DEMOKRITOS*, National Center for Scientific Research, France, 2004.

[11] N. Fitriah, N. Wahind, S. Kasim, and F. Hafit, "Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets," *IOP Publishing*, vol. 226, no. 1, 2017.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. .1, pp. 10–18, 2009.

[13] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive Bayes algorithm for spam filtering," in *Proceedings of the 2016 IEEE 35th International Performance Computing and Communications Conference*, pp. 1–8, IEEE, Las Vegas, NV, USA, December 2016.

[14] V. Vishagini and A. K. Rajan, "An improved spam detection method with weighted support vector machine," in *Proceeding of the 2018 International Conference on Data Science and Engineering*, pp. 1–5, IEEE, Kochi, India, August 2018.

[15] R. Karthika and P. J. W. T. C. Visalakshi, "A hybrid ACO based feature selection method for email spam classification," *WSEAS Transactions on Computers*, vol. 14, pp. 171–177, 2015.

[16] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Classifying phishing email using machine learning and deep learning," in *Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services*, pp. 1-2, IEEE, Oxford, UK, June 2019.

[17] S. Seth and S. Biswas, "Multimodal spam classification using deep learning techniques," in *Proceedings of the 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems*, pp. 346–349, IEEE, Jaipur, India, December 2017.

[18] C. Y. Li and L. Zheng, "Analysis of tai chi ideological and political course in university based on big data and graph neural networks," *Scientific Programming*, vol. 2021, Article ID 9914908, 9 pages, 2021.

[19] W. Fan, Y. Ma, Q. Li et al., "Graph neural networks for social recommendation," in *Proceedings of The World Wide Web Conference*, pp. 417–426, New York, NY, USA, May 2019.

[20] Y. Li, R. Jin, and Y. Luo, "Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (Seg-GCRNs)," *Journal of the American Medical Informatics Association*, vol. 26, no. 3, pp. 262–268, 2019.

[21] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an, "Graph convolutional encoders for syntax-aware neural machine translation," 2017, https://arxiv.org/abs/1704.04675.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[23] A. Trotman, A. Puurula, and B. Burgess, "Improvements to BM25 and language models examined," in *Proceedings of the 2014 Australasian Document Computing Symposium*, pp. 58–65, New York, NY,USA, November 2014.

[24] J. Zhou, G. Cui, S. Hu et al., "Graph neural networks: a review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[26] S. Douzi, F. A. AlShahwan, M. Lemoudden, and B. E. Ouahidi, "Hybrid email spam detection model using artificial intelligence," *International Journal of Machine Learning and Computing*, vol. 10, no. 2, pp. 316–322, 2020.

[27] D. Saini and M. Meena, "Hybrid forecasting scheme for enhance prediction accuracy of Spambase dataset," in *Proceedings of International Conference on Communication and Computational Technologies Algorithms for Intelligent Systems*, pp. 269–277, Springer, Singapore, August 2021.

[28] J. h. Kim and O. R. Jeong, "Knowledge graph-based Korean new words detection mechanism for spam filtering," *Journal of Internet Computing and Services*, vol. 21, no. 1, pp. 79–85, 2020.

[29] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, Doha, Qatar, January 2014.