IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2024, accepted 4 November 2024, date of publication 11 November 2024, date of current version 20 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3495830

RESEARCH ARTICLE

Advanced Analysis of Learning-Based Spam Email Filtering Methods Based on Feature Distribution Differences of Dataset

JIN-SEONG KIM, HAN-JIN LEE, HAN-JU LEE, AND SEOK-HWAN CHOI
Division of Software, Yonsei University, Wonju-si, Gangwon-do 26493, Republic of Korea

Corresponding author: Seok-Hwan Choi (sh.choi@yonsei.ac.kr)

This work was supported in part by the Ministry of Science and Information and Communication Technology (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2024-RS-2023-00259967; and in part by the 'Regional Innovation Strategy (RIS)' through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE), in 2024, under Grant 2022RIS-005.

ABSTRACT Spam emails, which are unsolicited bulk emails, pose a significant threat in digital communication security. To counter spam emails, learning-based spam email filtering methods have been extensively studied. However, as spam patterns evolve, these methods face challenges in maintaining the accuracy of models trained on outdated patterns. To demonstrate these limitations empirically and gain insight into the classification patterns of spam email filtering models, we propose an advanced analysis method to analyze the performance degradation of spam email filtering models. The proposed analysis method involves text preprocessing, embedding model training, spam email filtering model training, evaluation, and analysis of the classification patterns of the learning-based spam email filtering models. From the experimental results under various datasets and spam email filtering models, we show that the accuracy of spam email filtering models significantly decreases when the feature distribution of the test dataset is different from the training dataset. We also provides valuable insights for improving the model architecture, dataset structure, and training strategies by analysis of various factors such as confusion matrix, performance metrics, mean sequence length, out-of-vocabulary (OOV) rate, and top-20 tokens.

INDEX TERMS Spam email filtering, recurrent neural network (RNN), gated recurrent unit (GRU), long short-term memory (LSTM), ALBERT, security.

I. INTRODUCTION

Email is widely used as a key component of digital communication due to its flexibility and versatility, and it serves various purposes such as personal communication, sending work-related documents, and telemedicine. With the increase in email utilization, the volume of data has also surged. Consequently, effective management and storage of such large-scale data have emerged as significant challenges [1]. To address these issues, email service providers have adopted advanced digital technologies such as cloud computing, which enables users to store and access large volumes of emails. These advancements have made email a more practical and accessible means of communication.

However, the universal accessibility of email also makes it a common attack vector for malicious activities. Phishing attacks or malicious code transmissions via email can lead to attempts to steal users' cloud account information, compromising sensitive data. Typically, such attacks are carried out through unsolicited emails known as spam. Spam emails are sent in bulk to numerous unspecified recipients and are often designed to mimic legitimate emails, making them difficult to filter using traditional text processing methods. To tackle these problems, learning-based approaches have been introduced into spam email filtering. Early learning-based approaches used machine learning models, such as Logistic Regression(LR), Naive Bayes(NB), and Support Vector Machine to filter large amounts of spam emails or to improve accuracy of spam email filtering [2], [3], [4], [5]. Since then, with the emergence of deep learning models, learning-based approaches have opened new

The associate editor coordinating the review of this manuscript and approving it for publication was Ángel F. García-Fernández.

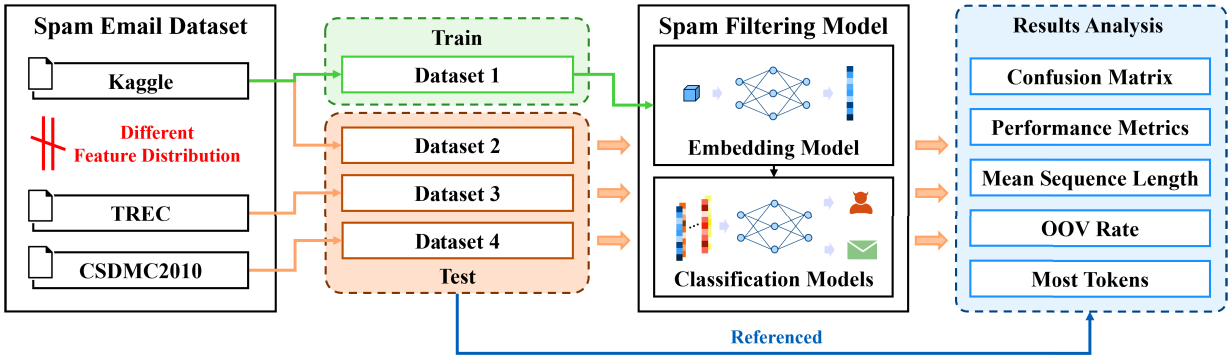

FIGURE 1. Overall procedure of our analysis method.

avenues for spam email filtering. Deep learning models have dramatically improved the accuracy of spam email filtering methods by handling extensive amounts of data [6], [7], [8], [9]. Furthermore, deep learning-based natural language processing techniques have played an important role in interpreting the meaning and context of emails. By introducing such learning-based analysis methods, many studies have demonstrated high accuracy and effectiveness in spam email filtering tasks.

Despite these efforts, spam emails remain a persistent challenge for many applications due to the evolving nature of spam patterns. For example, patterns from early spam emails, such as specific keywords or HTML layouts, can eventually appear in legitimate emails over time. Furthermore, spammers continuously develop strategies to disguise their messages as legitimate emails by exploiting the complexity, diversity, and ambiguity of natural language. This evolution causes a decline in the accuracy of spam email filtering models trained on outdated feature distributions.

Offer new insights into learning-based email spam filtering, in this paper, we introduce an advanced method for analyzing the classification patterns of deep learning-based spam email filtering models. Our approach aims to address the limitations of traditional learning-based models. At each step of the proposed analysis method, we performed the following procedures: First, we performed data preprocessing for email data using datasets with three different feature distributions. This includes tokenization, text normalization, and text cleaning. Second, we trained four learning-based spam email filtering models (RNN, GRU, LSTM and ALBERT) using the preprocessed email dataset. Third, we obtained the classification results for each model using three different test datasets that reflect changes in spam email patterns over time. Finally, we performed an advanced analysis of classification results based on various factors, i.e., confusion matrix, mean sequence length, out-of-vocabulary (OOV) rate, and most frequently appearing tokens. A visual summary of each step is shown in Fig. 1.

From the experimental results across various spam email filtering datasets and model configurations, we gained key insights into the behavior of learning-based spam filtering models and the characteristics of the data they learn from. These insights can guide the design of model architectures, dataset structures, and learning strategies. Additionally, applying these findings can improve existing spam filtering methods, making them more accurate and adaptable to evolving spam patterns.

The rest of the paper is organized as follows. In Section II, we provide an overview of well-known machine learning-based and deep learning-based spam email filtering methods. Also, we describe the motivation of this paper. In Section III, we describe the threat model, the overall operation, and the details of the proposed analysis method. In Section IV, we show the analysis results from various factors under different models and different datasets. In Section V, we discussed our comprehensive evaluation of the classification results. Finally, we conclude this paper in Section VI.

II. RELATED WORKS

In this section, we provide an overview of learning-based approaches for spam email filtering methods, which have been widely referenced in many works. We also explain the motivation behind the proposed analysis method by highlighting the limitations of existing learning-based spam email filtering methods.

A. MACHINE LEARNING BASED METHODS

In this section, we introduce different machine learning methods, which were proposed to filter spam email.

Machine learning approaches, such as logistic regression and support vector machines, are frequently used for spam email filtering due to their simplicity and efficiency. Several studies have evaluated the effectiveness and superiority of various machine learning algorithms in spam email filtering [10], [11], [12], [13]. Furthermore, some studies have introduced different learning strategies to improve the accuracy of machine learning-based spam email filtering [14], [15], [16], [17].

Gibson et al. proposed a method to filter spam emails based on a machine learning model with bio-inspired metaheuristic algorithms [14]. They applied Particle Swarm Optimization and Genetic Algorithms to improve the spam filtering accuracy of five machine learning algorithms, i.e., Naive Bayes, Support Vector Machine, Random Forest,

Decision Tree, and Multi-Layer Perceptron. Dedeturk and Akay proposed logistic regression trained with the artificial bee colony(ABC) algorithm for spam email filtering [15]. The ABC algorithm optimizes parameter combinations by modeling the global search process using three types of artificial bees: hired bees, onlooker bees, and scout bees. Feng et al. proposed an improved spam email filtering method using the SVM-NB algorithm [16]. By eliminating bad samples with SVM and training a Naive Bayes-based classifier, the SVM-NB algorithm enhances both effectiveness and efficiency in filtering spam email. Omotehinwa and Oyewola proposed hyperparameter tuning techniques of Random Forest and extreme gradient boost (XGBoost) for filtering spam email [17]. To find optimal hyperparameter values, they used a 10-fold cross-validation technique and grid search technique.

B. DEEP LEARNING BASED METHODS

In this section, we introduce advanced deep learning techniques for spam email filtering, such as RNN, GRU, LSTM, and BERT. These methods have been shown to outperform traditional machine learning approaches in filtering accuracy [9], [18], [19], [20], [21], [22]. Specifically, their ability to capture complex sentence relationships and efficiently process large datasets has led researchers to explore enhanced approaches for improved spam email filtering accuracy [23], [24], [25], [26], [27].

Yang et al. introduced a multi-modal architecture based on model fusion (MMA-MF) in order to achieve higher accuracy than traditional spam email filtering methods [23]. Specifically, they combined CNN and LSTM to consider both text and image features in spam emails. Zavrak and Yilmaz introduced a hierarchical approach to extract generalizable, abstract, and meaningful features from spam email [24]. The hierarchical approach consists of two main layers and is implemented using FastText (FT) and Hierarchical Attentional Hybrid Neural Networks (HAN), respectively. Chen et al. proposed a method to reduce labeling cost and improve model adaptability in LSTM-based spam email filtering method [25]. They sampled only the most valuable data for training the email spam filtering model by using Least Confidence(LC) and Max Entropy(ME). Khan et al. proposed a performance measurement method to evaluate spam email filtering methods using fuzzy logic [26]. Specifically, they combined Unified And-Or (UAO) logic with accuracy, recall, and precision to create a new evaluation metric based on fuzzy logic concepts. Abdal et al. proposed a method for spam email filtering by leveraging the fine-tuning of the pre-trained ALBERT model, an optimized and computationally efficient variant of BERT [28]. Their approach showed strong performance across multiple datasets, highlighting the model's efficiency and effectiveness.

C. LIMITATION OF PREVIOUS LEARNING-BASED SPAM EMAIL FILTERING METHODS

Learning-based spam email filtering methods have shown their effectiveness in spam email filtering by utilizing various machine learning and deep learning models. However, the performance of these methods depends significantly on the feature distribution of the training dataset. Specifically, when the feature distribution of the test dataset differs from the training dataset, the model's filtering performance decreases, leading to more misclassifications. For example, a learning-based spam email filtering model trained on the "Kaggle Email Spam" dataset showed low filtering performance when tested on the "CSDMC2010 Spam Corpus" dataset [29]. This is due to the shift in feature distributions between the datasets, as each dataset captures different spam patterns and reflects language changes over time.

These observations motivate us to conduct a more in-depth analysis of learning-based spam email filtering methods, particularly in dynamic environments where the feature distributions between training and test datasets shift. Also, to further evaluate the generalization performance of spam email filtering models under different conditions, we assessed models such as RNN, LSTM, GRU, and ALBERT with varying model sizes and their ability to handle distribution shifts.

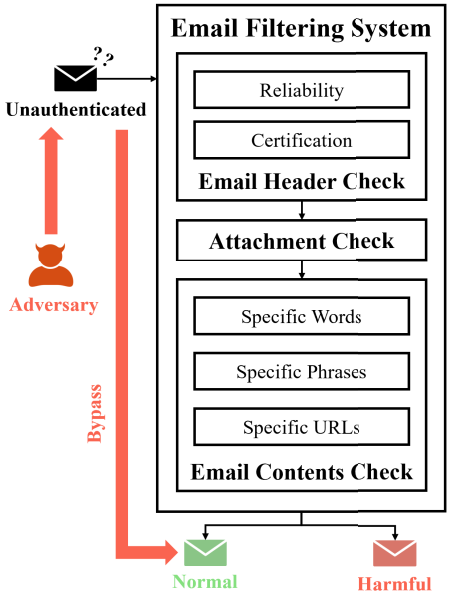

FIGURE 2. Threat model: the normal flow of spam email filtering system vs. Adversarial flows that bypass spam emails.

III. PROPOSED METHODS

In this section, after outlining the targeted threat model, we introduce a proposed analysis method to gain new insights into the functional complement of learning-based email spam filtering.

A. THREAT MODEL

As an example of an email filtering service, let us consider a learning-based spam email filtering system that uses only the content of emails. Here, we assume that the architects of the spam email filtering system cannot utilize email headers

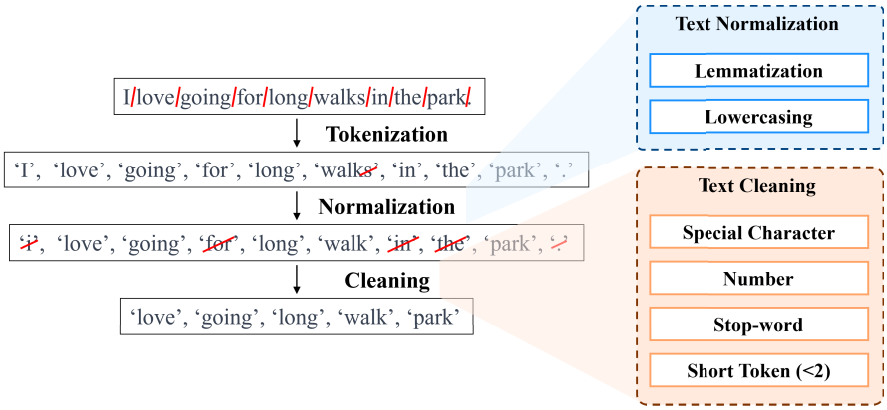

FIGURE 3. Overall procedure of preprocessing.

and rely solely on email content for spam email filtering. Also, we assume that the spam email filtering systems use learning-based models such as RNN, GRU, LSTM and ALBERT to classify spam emails. Now, let us consider an adversary who aims to distribute harmful information and malicious code to email users via spam email. To generate malicious spam emails, the adversary uses various contents designed to evade traditional spam filtering systems. In other words, each spam email content contains new patterns that were not used in previous spam emails. For example, the adversary can generate spam emails that reflect recent news or issues, use novel styles or language, or make them look like legitimate emails.

In this threat model, the goal of the spam email filtering method is to quickly adapt to evolving spam email patterns. To achieve this, the spam email filtering method must continuously update the pre-trained spam filtering model by analyzing classification results from newly collected datasets, allowing it to learn and detect emerging spam trends.

B. OVERALL OPERATION

In this section, we describe the overall process for advanced analysis of classification patterns in learning-based spam email filtering models. To conduct this advanced analysis, we implemented the following 4 procedures: First, we performed data preprocessing to convert spam emails into vector forms which the spam email filtering model can process. Specifically, in this procedure, we transformed text in spam email datasets into a consistent form using word tokenization, text normalization, and text cleaning. Second, we trained a word embedding model to reduce the dimensionality of large spam email data while effectively capturing the semantic relationships between words. Third, we trained a learning-based spam email filtering model to categorize emails into spam and legitimate emails. Finally, we analyzed the classification behavior of the learning-based spam filtering model using key metrics like the confusion matrix and frequent words in misclassified emails, gaining valuable insights for future training strategies. In Fig. 1, we show the overall procedure of the proposed analysis method.

C. TEXT PREPROCESSING

In this section, we describe the text preprocessing procedure for spam email data used in the proposed analysis method. Emails often contain many unnecessary elements to interpret the meaning of the sentence, such as stop words and special characters. They also contain words in various forms, such as plural nouns, verb tense changes, and adverbs. This not only increases the amount of training data but also hinders the proper learning of spam email filtering models. Thus, it is important to remove unnecessary components of email and unify the format for effective spam email filtering. In the proposed analysis method, text preprocessing is performed with three steps: tokenization, normalization, and cleaning, as shown in Fig. 3.

1) TOKENIZATION

In this step, we tokenized emails to separate paragraphs or sentences into small analytical units. After tokenization, the emails are divided into tokens, representing the smallest units of analysis. Specifically, we tokenized the email by word-wise separation using TreeBank tokenization technique. The TreeBank tokenization technique is performed as follows:

- Words consisting of hyphens (-) are treated as a single token.(e.g., real-time, state-of-the-art, COVID-19, self-respect)
- Words with apostrophes are separated into separate tokens. (e.g., "isn't" –> "is", "n't", "can't" –> "ca", "n't")
- Punctuation is treated as a single token with the preceding or following word, unless it appears at the end of a sentence.(e.g., "3.88", "Ph.D.", "A.M.", "Jr.", "Inc.".)

2) TEXT NORMALIZATION

In this step, we normalized tokenized words to convert various forms of words into a consistent form. In emails written in natural language, even words with the same meaning often appear in various forms. For example, words like "play", "played", and "playing" have similar meanings,

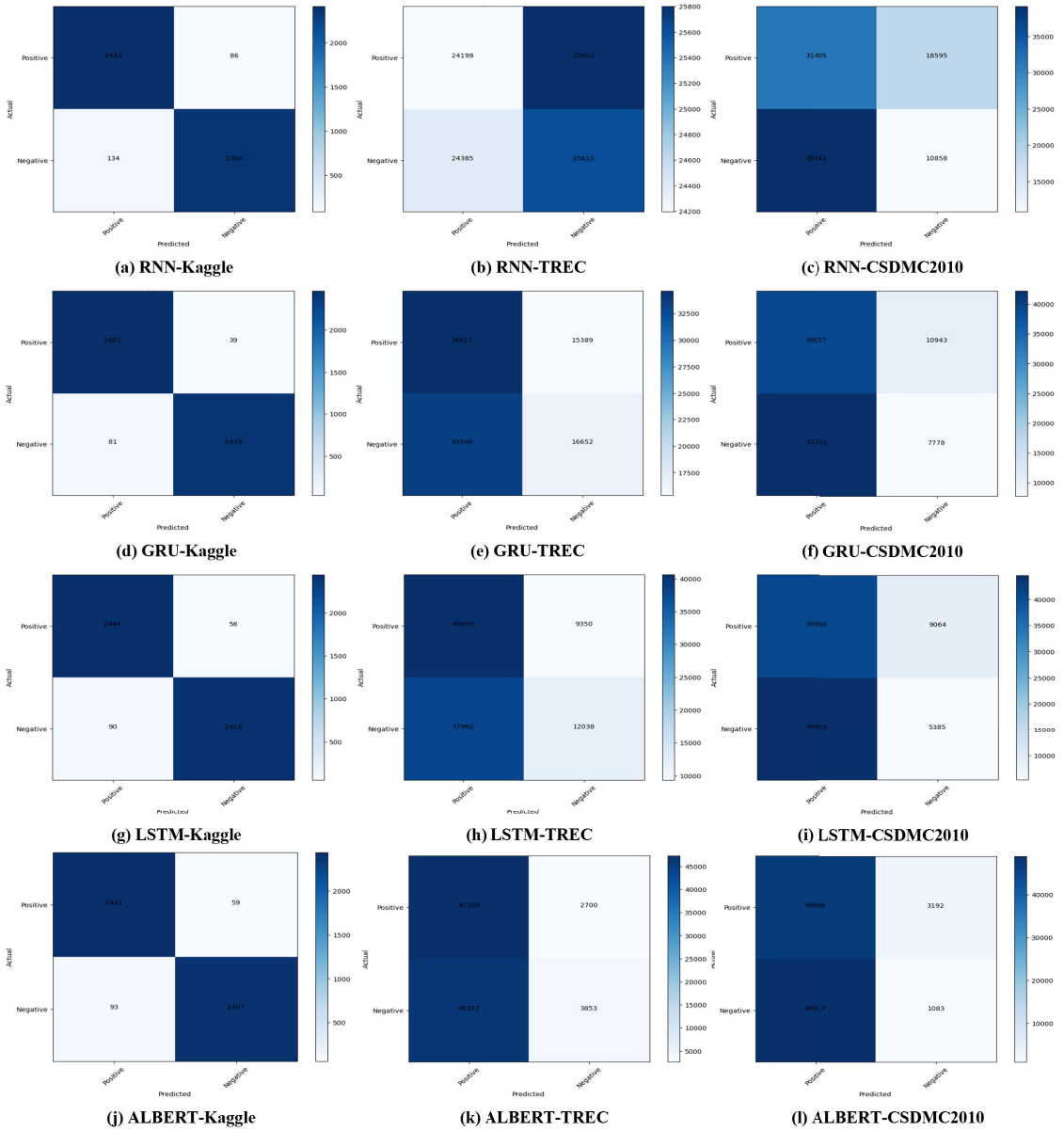

FIGURE 4. Confusion matrix for classification results by spam email datasets and filtering models.

but are expressed differently. such variations increase data sparsity, which in turn raises processing time and resource consumption in data-driven analysis methods. Also, this makes it difficult to capture the exact meaning of a sentence or document, which can reduce the accuracy of spam email filtering models. Thus, we normalized the form of the tokens to minimize these negative effects. Specifically, we converted the text to lowercase and performed lemmatization to keep the tokens consistent.

3) TEXT CLEANING

In this step, we performed text cleaning to remove information that acts as unnecessary noise in spam email filtering. Specifically, text cleaning involved four removal tasks:

- **Special characters Removal:** Special characters contain text components including emoticons, punctuation marks, and mathematical symbols. We removed such special characters because they often act as noise in spam email filtering.
- **Number Removal:** Numbers generally increase computational complexity or act as noise in text analysis. Therefore, we removed tokens consisting of numbers to reduce the computation and improve data quality.
- **Stop-word Removal:** Stop words are words such as ‘‘a’’, ‘‘the’’, and ‘‘but’’, which often appear in the text data, but do not contribute to analysis. We removed these stop words because they increase the computational load of the email filtering model and reduce its efficiency.

- **Short Token Removal:** Short tokens usually have no special meaning and can interfere with the training process of spam email filtering models. Thus, we removed tokens with a length of 2 or less.

D. LEARNING WORD EMBEDDING MODEL

In this section, we describe a word embedding model for representing emails as vectors. Embedding is a technique that converts text data into numerical vector at the level of words, sentences, documents, etc. These vectors transformed by embedding are used in various text analysis tasks. Since the spam email filtering model learns spam email patterns through input in vector form, embedding is essential for spam email classification.

In this paper, we used the Word2Vec model which is commonly referenced in many studies for spam email filtering model. Word2Vec is a learning-based word embedding model that can represent words as vectors by considering the context of surrounding words. Specifically, we embedded emails using the skip-gram scheme, which embeds words by predicting surrounding words based on the central word. The skip-gram method predicts the central word ''cat'' by considering the surrounding words ''The'', ''sat'', ''on'', and ''mat'' in the sentence ''The cat on the mat''. This method can effectively capture complex semantic relationships between words in an email. Furthermore, vectors learned through the skip-gram method simplify dense representations and reflect subtle differences in word usage patterns between spam and legitimate emails. As a result, we converted the email into a 64-dimensional embedding vector in this step. In addition, to examine whether similar trends are observed in more recent and larger models, we adopted ALBERT's embedding pre-training approach [30]. Specifically, we pre-trained ALBERT with an embedding size of 128, leveraging key techniques such as the Sentence-Order Prediction (SOP) task, Masked Language Modeling (MLM), Cross-Layer Parameter Sharing, and WordPiece Tokenization.

E. LEARNING CLASSIFICATION MODEL

The RNN model is a simple iterative connection that provides sequence learning for spam emails and forms the foundation of our approach. However, RNNs have a problem of long-term dependencies due to their structural limitations, leading to reduced effectiveness for long input sequences. To address these limitations, GRU and LSTM were introduced. Specifically, LSTM incorporates a sophisticated gating mechanism to regulate the flow of information, making it highly effective for tasks that require retaining information over extended sequences. GRUs simplify the LSTM architecture by combining the forget and input gates into a single update gate, reducing the complexity of the model while maintaining the ability to manage long-term dependencies. In addition, we utilized ALBERT, a lightweight and efficient version of BERT. Finally, to further improve the spam email filtering performance, we optimized the model by using a single-layer architecture with 32-dimensional hidden vectors for RNN-based models and 128-dimensional hidden vectors for ALBERT.

F. ANALYZING CLASSIFICATION PATTERNS

To gain insights into improving spam email filtering, we analyzed the classification patterns of various models. We identified key features influencing their ability to distinguish between spam and legitimate emails by evaluating metrics like accuracy, precision, and recall. Additionally, we examined text characteristics such as word frequency and token distribution to better understand the training and testing data.

The reasons for analyzing classification patterns in spam email filtering models are as follows:

- **Accurate evaluation of spam email filtering**: By using a confusion matrix and performance metrics, such as accuracy, recall, precision, and F1-score, we can accurately assess the overall performance of each spam email filtering model. We can also identify failure cases of spam email filtering.
- **Identification of the cause of performance degradation**: Analysis of classification patterns helps to identify factors that reduce the performance of spam email filtering models. For example, by analyzing mean sequence length and OOV rate of the input data, we can understand how the performance of spam email filtering model varies with sentence length or words that are not learned.
- **The proposition of strategies for performance improvement**: From the analysis results, we can extract information about tokens with a high frequency of misclassification. By using this information, we can suggest strategies for improving the performance of spam email filtering models.

To effectively analyze the performance of spam email filtering models, we used mean sequence length and OOV rate as evaluation metrics. Mean sequence length represents the average length of input sentences. By analyzing true and false cases, we can identify where the email spam filtering model has difficulty distinguishing based on sentence length. The OOV rate represents the ratio of words in the input sentence that were not part of the training data. By analyzing TN, TP, FN, and FP, we can identify cases where the model has difficulty processing OOV words.

Also, we measured accuracy, recall, precision, and F1-score to evaluate the performance of spam email filtering models. Each metric can be calculated as follows:

$$Accruacy = \frac{TP + TN}{N}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

TABLE 1. Performance metrics for classification results by spam email datasets and filtering models.

Dataset	Model	Performance Metrics			
		Accuracy	Recall	Precision	F1-Score
Kaggle	RNN	0.91	0.94	0.88	0.91
	GRU	0.94	0.95	0.92	0.93
	LSTM	0.92	0.94	0.9	0.92
	ALBERT	0.97	0.96	0.98	0.97
TREC	RNN	0.54	0.52	0.88	0.66
	GRU	0.64	0.6	0.8	0.69
	LSTM	0.56	0.54	0.88	0.67
	ALBERT	0.51	0.51	0.95	0.66
CSDMC2010	RNN	0.47	0.48	0.84	0.61
	GRU	0.41	0.44	0.64	0.52
	LSTM	0.47	0.49	0.85	0.62
	ALBERT	0.48	0.49	0.94	0.64

Here, TP represents the number of true positives, FP represents the number of false positives, FN represents the number of false negatives and TN represents the number of true negatives.

IV. EXPERIMENTAL RESULTS

To show how effective the proposed analysis method is when applied to real-world scenarios, we measured the classification results of learning-based spam email filtering models under various factors, including performance metrics.

A. EXPERIMENTAL ENVIRONMENT

In this experiment, we used datasets generated from various online sources and contests to analyze classification patterns in spam email filtering models.

These datasets, which include email headers, subjects, and body text, were collected to provide a comprehensive view of classification patterns for learning-based spam email filtering models across various sources and time periods. Each dataset contains emails collected at varying times and from different origins, allowing us to capture evolving trends in spam emails. Furthermore, the diversity in email sources, creation dates, and linguistic characteristics enables us to evaluate the model's generalization capability and performance across a wide range of spam email types.

After training a filtering model using Kaggle's spam email dataset, we assessed its performance on misclassification tendencies by testing it on two other datasets: the TREC Public Spam Corpus and the CSDMC 2010 SPAM corpus. Kaggle's Spam Mails Dataset consists of 5,171 emails with subjects and contents, categorized into spam and non-spam (ham). To create a more practical evaluation environment, we also used the TREC Public Spam Corpus and the CSDMC 2010 SPAM corpus. The TREC Public Spam Corpus is a widely-used dataset for practical spam email classification, containing 75,419 emails categorized into two classes. The CSDMC 2010 SPAM corpus includes 3,837 emails, also categorized into spam and ham. Specifically, for the Kaggle dataset, we prepared 10 test sets, each containing 500 emails divided evenly between 250 spam and 250 non-spam emails. For both the TREC and CSDMC2010 datasets, we arranged 10 test sets comprising 1,000 emails each, with an equal split of 500 spam and 500 non-spam emails. This balanced test configuration allows for a fair assessment of the model's performance, providing insights into its real-world applicability and highlighting specific patterns in classification. This approach ensures a comprehensive evaluation of the model's effectiveness and helps identify areas for improvement in spam email filtering technologies.

In this experiment, the embedding model contains a total of 42,579 tokens, and each token is embedded in 64 dimensions when using Word2Vec. For ALBERT, the model contains 29,185 tokens, and each token is embedded in 128 dimensions. The spam filtering models take the embedding vector of each token as input and perform feature extraction and classification through a recurrent neural network (RNN) model. Specifically, the RNN model receives a 64-dimensional embedding vector and outputs a 32-dimensional hidden vector. This hidden vector is then passed through a fully connected layer to obtain the final classification result. Similarly, the GRU and LSTM models receive 64-dimensional embedding vectors, output 32-dimensional hidden vectors, and perform final classification through a fully connected layer. For ALBERT, the model utilizes a 128-dimensional embedding vector, allowing it to capture more complex patterns and dependencies in the data, which is particularly beneficial for spam filtering tasks.

In model training, the same embedding model was used for RNN, GRU, and LSTM, while ALBERT employed its own pre-trained embeddings. To ensure diversity among the spam email filtering models, a total of 40 different architectures—10 for each model type (RNN, GRU, LSTM, ALBERT)—were randomly trained. The diversity of learning models facilitated an assessment of their generalization and adaptability across different datasets, providing valuable insights into their performance.

We implemented the word embedding model and spam classification model using Pytorch 1.7.1(+cu110) and Python version 3.9.16 and performed text preprocessing by using the NLTK library. For the efficient experiments, we performed experiments on the Windows 10 Home Edition machine with build version 22H2, 3.00GHz CPU clock(13th Gen Intel Core CPU I9-13900K), 8,704 GPU cores(RTX3080), and 32GB memory.

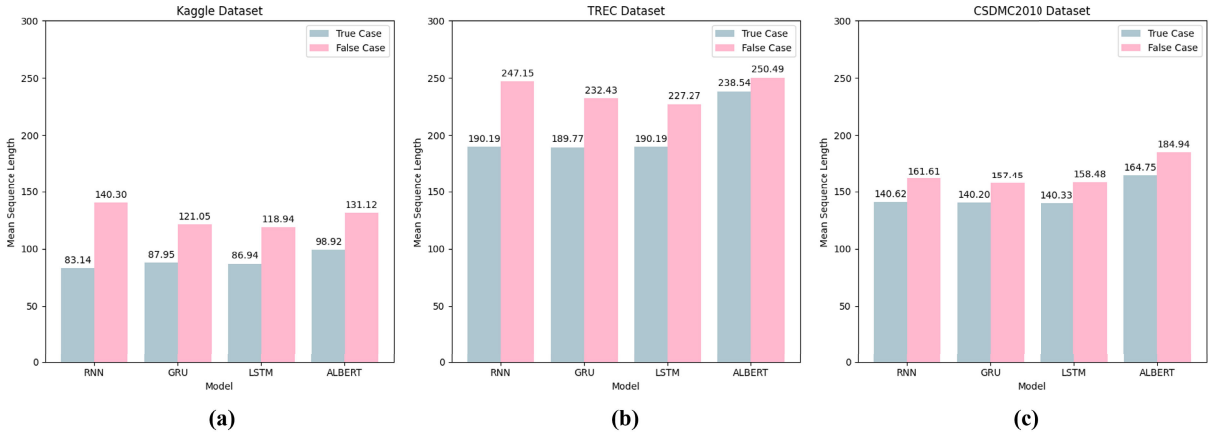

FIGURE 5. Mean sequence length for classification results by spam email datasets and filtering models.

B. EXPERIMENTAL ANALYSIS

1) CONFUSION MATRIX

To analyze errors and classification patterns in spam email filtering from various aspects, we measured a confusion matrices from different spam email filtering models and datasets.

In Fig. 4, we show the confusion matrix for the RNN, GRU, LTSM and ALBERT models, trained on the Kaggle dataset and evaluated on the Kaggle dataset, the TREC Dataset, and the CSDMC2010 Dataset, respectively. We observed that spam email filtering models tend to classify more emails as spam when the feature distributions of the training and the test datasets are different. For example, while the LSTM model evaluated on the Kaggle dataset classified 2,534 out of 5,000 emails (50.68%) as spam, the LSTM model evaluated on the TREC Dataset classified 78,612 out of 100,000 emails (78.61%). This tendency was even stronger on the CSDMC2010 dataset compared to the TREC dataset. For example, while the GRU model evaluated on the TREC dataset classified 67,959 out of 100,000 emails (67.95%) as spam, the GRU model evaluated on the CSDMC2010 dataset classified 81,279 out of 100,000 emails (81.27%) as spam. This is because the CSDMC2010 dataset has a similar feature distribution to spam emails in the Kaggle dataset than to the TREC dataset.

2) PERFORMANCE METRICS

To evaluate the effectiveness of RNN, GRU, LTSM and ALBERT in spam email filtering, we measured various performance metrics such as accuracy, recall, precision, and F1-score. In Table 1, we show the spam email filtering performance of the RNN, GRU, LTSM and ALBERT models, trained on the Kaggle dataset and evaluated on the Kaggle dataset, the TREC Dataset, and the CSDMC2010 Dataset, respectively. As shown in the 'Kaggle' row in Table 1, all spam email filtering models showed good performance on the Kaggle dataset. For example, the ALBERT model evaluated on the Kaggle dataset showed 0.97, 0.96, 0.98, and 0.97 for the accuracy, recall, precision, and F1-score, respectively.

Moreover, as shown in the 'precision' column in Table 1, the spam filtering models maintained high precision in classifying actual spam emails, despite difference in feature distribution between the training and testing datasets. For example, the GRU and LSTM models evaluated on the Kaggle dataset showed 0.92 and 0.9 for precision, respectively. On the other hand, as shown in the 'TREC' and 'CSDMC2010' rows in Table 1, the spam filtering models showed low classification performance for legitimate emails when the feature distribution of the test dataset differs from the training dataset. In particular, in this case, as shown in the 'recall' column of Table 1, all spam filtering models showed significantly lower recall compared to precision. This suggests that spam filtering models have the potential to misclassify legitimate or important emails as spam.

3) MEAN SEQUENCE LENGTH

To analyze the impact of sequence length on spam email filtering performance, we measured the mean sequence length of the input data of the spam email filtering model, separately as correctly classified and incorrectly classified.

In Fig. 5, we show the mean sequence length of input emails for the RNN, GRU, LSTM and ALBERT models. Each model was trained on the Kaggle dataset and evaluated on the Kaggle dataset, the TREC Dataset, and the CSDMC2010 Dataset, respectively. As shown in the comparison of 'True Case' and 'False Case' in Fig. 5, the mean sequence length of emails incorrectly classified by the spam email filtering model was longer than that of emails correctly classified. For example, while the RNN model evaluated on the Kaggle dataset showed 83.14 mean sequence length for True cases, it showed 140.3 mean sequence length for False cases. This suggests that the model is more likely to make errors when processing longer sequences.

Additionally, the analysis reveals that longer emails, as seen in the TREC dataset, pose greater challenges for spam filtering models, leading to higher misclassification rates. In contrast, shorter emails, like those in the Kaggle dataset,

TABLE 2. Out-of-vocabulary (OOV) rates for classification results by spam email datasets and filtering models.

Dataset	Model	OOV Rate			
		TP	FP	TN	FN
TREC	RNN	0.424	0.525	0.302	0.319
	GRU	0.424	0.545	0.313	0.323
	LSTM	0.424	0.663	0.303	0.315
CSDMC	RNN	0.211	0.187	0.171	0.173
	GRU	0.211	0.212	0.171	0.167
	LSTM	0.212	0.215	0.171	0.17

are easier for models to classify correctly. The CSDMC2010 dataset, with its balanced sequence lengths, provides a useful benchmark for evaluating model performance across varying email lengths. These findings suggest that enhancing model performance on longer sequences could improve overall spam filtering effectiveness, and incorporating datasets with diverse sequence lengths is essential for comprehensive model evaluation.

4) OOV RATE

To analyze the tendency of the spam email filtering models according to the ratio of OOV tokens, we measured the ratio of OOV tokens divided into True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP). In this experiment, we excluded evaluations on the Kaggle dataset because the data used for learning does not contain enough tokens to make the analysis meaningful. Additionally, ALBERT was excluded from this experiment as its encoding and vocabulary generation methods significantly reduce the occurrence of OOV tokens, making it less relevant for this analysis.

In Table 2, we show the percentage of OOV tokens for the RNN, GRU, and LSTM models, trained on the Kaggle dataset and evaluated on the TREC Dataset, and the CSDMC2010 Dataset, respectively. As shown in the 'FP' column in Table 2, we observed that the OOV rate tends to increase as the complexity of the model increases for emails classified as False Positive. For example, under the TREC dataset, the RNN model showed 0.525 OOV rate for emails classified as False Positive, while the LSTM model showed 0.663 OOV rate. This suggests that more complex models are more likely to misclassify emails containing words not in the training data as spam. However, as shown in the 'CSDMC2010' row in Tabel 2, the OOV rates for True Positive and False Positive are relatively low and similar across all models. Likewise, the OOV rates for True Negative and False Negative are also similar, showing consistent results for all models. This suggests that the CSDMC2010 dataset is lexically more uniform compared to the TREC dataset.

5) TERM FREQUENCY IN ACROSS VARIOUS EMAIL FILTERING MODELS

To analyze the impact of term frequency on spam email filtering performance, we measured the tokens that appear most frequently in the classification results, divided into true and false cases. In this experiment, we categorized tokens into terms related to the web, commerce, general communication, business, style, and subwords.

Table 3 shows the tokens that appear most frequently in input emails where the spam email filtering models (RNN, GRU, LSTM and ALBERT) classified spam email datasets (Kaggle, TREC, and CSDMC2010) as spam. As shown in the 'True Positive' column in Table 3, we observed that web-related terms and commercial terms were frequently used in emails when the spam filtering models correctly classified input emails as spam. This demonstrates that such tokens act as key elements in spam email filtering models. However, as shown in the 'False Positive' column in Table 3, we observed that spam emails frequently used not only web-related and commercial terms but also business and styling terms when spam filtering models failed to classify spam. This is because spam emails have become sophisticated enough over time to bypass spam email filtering models by mimicking legitimate emails. For example, the CSDMC2010 dataset, generated in 2010, contains more business terms in its spam emails, such as 'user', 'problem', 'work', 'system', 'file', 'unsubscribed', and 'list', than the Kaggle dataset, which is based on the Enron dataset generated in 2002.

Furthermore, the tokens listed in the 'True Negative' column in Table 3 highlight that legitimate emails often contain more varied and context-specific terms, which differ significantly from the repetitive and predictable patterns found in spam emails. For example, tokens that frequently appear in emails successfully classified as legitimate emails on the Kaggle dataset include varied commercial terms such as 'enron', 'gas', and 'hpl' and general communication terms such as 'thanks' and 'need'. Similarly, as shown in the 'False Negative' column of Table 3, e-mails that the spam email filtering model incorrectly classified as spam often feature various words, including common communication terms. This suggests that spam emails increasingly try to avoid detection using normal communicative representations.

In addition, the analysis of Special/Subword tokens in Table 3 suggests that the presence of a higher number of shorter subword tokens (marked by ##) compared to essential special tokens like '[CLS]' and '[SEP]' may indicate less meaningful data for spam filtering. When a model frequently encounters fragmented subwords, it is more likely dealing with altered or incomplete words, potentially reducing the richness of contextual information available for accurate classification. This could mean that spam emails employing such tactics may be more difficult to detect accurately, as they attempt to bypass filters by fragmenting or distorting key spam-related terms. Thus, a higher ratio of subword tokens could reflect an increase in noise, making it more challenging for the model to effectively distinguish between spam and legitimate emails.

These observations indicate that while spam filtering models can effectively identify common spam characteristics, they may struggle to distinguish more sophisticated spam emails that mimic legitimate emails.

TABLE 3. Top-20 tokens for classification results by spam email datasets and filtering models.

Category	Dataset	Top 20 Tokens			
		TP	FP	TN	FN
Web-related	Kaggle	'com', 'http', ...	'com', 'http', ...	-	-
	TREC	'http', 'charset', 'online', ...	'com', 'http', 'www', ...	'http', 'email'	'http', 'email'
	CSDMC2010	'http', 'email'	'http', 'email'	'http', 'email'	'http', 'email'
Commercial	Kaggle	'pill', 'price', 'product', ...	-	'enron', 'gas', 'hpl' ...	-
	TREC	'pill', 'price', 'item', ...	-	-	-
	CSDMC2010	-	-	'click', 'free', 'price', ,,,	'free', 'click', 'price'
General Comm.	Kaggle	'please', 'new', 'get', ...	-	'thanks', 'need', ...	'please', 'get', 'believe', ...
	TREC	'know', 'said', 'see', ...	-	'list', 'new', 'please', ...	'new', 'list', 'story', ...
	CSDMC2010	'wrote', 'list', ...	'wrote', 'use', 'get', ...	'please', 'get', 'time', ...	'please', 'get', 'new', ...
Business	Kaggle	-	-	-	'enron', 'david'
	TREC	-	-	'code' , 'file'	'unsubscribe'
	CSDMC2010	'user'	'user', 'problem', ...	'business', 'state', ...	'business', 'information', ...
HTML/Styling	Kaggle	-	'font', 'height', ...	-	-
	TREC	-	'font', 'height', ...	-	-
	CSDMC2010	-	-	'font', 'color', 'nbsp'	'font', color', 'nbsp'
Special/Subword	Kaggle	'[CLS]', '[SEP]', '##e', ...	'##c', '[CLS]', '[SEP]', ...	'[CLS]', '[SEP]'	'[CLS]', '[SEP]', '##er', ...
	TREC	'##s', '##n', '##ty', ...	'##ja', '##ins', '##rd', ...	'##h', '##s', '##iness', ...	'##s', '##ing', '##or', ...
	CSDMC2010	'##s', '##ian', '##g', ...	'##s', '##ian', '##e', ...	'##s', '[CLS]', '[SEP]', ...	'##s', '[CLS]', '[SEP]', ...

V. DISCUSSION

In conclusion, our comprehensive evaluation of RNN, GRU, LSTM, and ALBERT models for spam email filtering provides significant insights into their performance across various datasets and models. The analysis highlights that these models demonstrate the following:

Firstly, the confusion matrix shows that spam email filtering models tend to classify emails one class when there is a significant difference in feature distribution between the training and testing datasets. This highlights the importance of aligning feature distributions to improve model performance, as significant discrepancies can lead to higher misclassification rates.

Secondly, while the models generally achieve high precision in identifying spam emails, they struggle with recall, particularly when feature distributions differ between the training and testing datasets. This leads to an increased vulnerability to false positives, where legitimate emails are misclassified as spam, indicating the need for models with more robust generalization capabilities across varying datasets.

Thirdly, mean sequence length analyses demonstrate that longer and more complex emails pose additional challenges for the models. The increased difficulty in processing these emails leads to a higher misclassification rate, suggesting that the models may be overfitting to shorter or less complex sequences encountered during training.

Fourthly, high out-of-vocabulary (OOV) rates in more complex models, such as LSTM, exacerbate these challenges, as unfamiliar tokens within emails increase the likelihood of misclassification. This issue highlights the limitations of certain models in handling diverse email structures, particularly when the training vocabulary does not fully capture the token distributions of the test data.

Finally, term frequency analysis reveals that spam emails are becoming increasingly sophisticated, often mimicking legitimate emails to evade detection. This trend highlights the need for more adaptable and robust spam email filtering models. Our advanced analysis of frequently occurring tokens in misclassified emails suggests a potential improvement strategy: by monitoring the frequency of subword tokens, we can identify areas where current models are vulnerable and provide a new metric to evaluate their robustness.

This approach can be leveraged to guide both the collection of more representative datasets and the training of more adaptable models. By focusing on the evolving nature of spam email content, we can enhance precision and adaptability, addressing the shortcomings identified in current models. Thus, this analysis not only deepens our understanding of spam email filtering model limitations but also outlines a clear path for refining future spam detection techniques.

VI. CONCLUSION

The proliferation of email usage has led to an increasing problem with spam emails, which often carry phishing content or malicious code. To solve these threats, two types of learning-based spam email filtering methods have actively been studied as an efficient defense method: (1) machine learning-based methods; and (2) deep learning-based methods. However, learning-based spam email filtering methods often suffer from reduced accuracy when there is a difference between the feature distributions of the training and test datasets. In this paper, we conducted an in-depth analysis of learning-based spam email filtering models. Our analysis used various metrics, including confusion matrices, performance metrics, mean sequence length, out-of-vocabulary (OOV) rates, and most frequently used tokens. From our experimental results, analyzed through various metrics, we provided valuable insights into the learned data characteristics of spam email filtering models. Additionally, such results can be actively used not only to design model architecture, dataset structure, and learning strategies for spam email filtering but also to improve existing spam email filtering methods.

REFERENCES

[1] *Email Threat Landscape Report: Protecting Your Organization From Increased Malware, BEC, and Credential Phishing Attacks*, Trend Micro, Tokyo, Japan, 2024.

[2] S. Prof and T. Verma, "Email spam detection and classification using SVM and feature extraction," *Int. J. Adv. Res., Ideas Innov. Technol.*, vol. 3, no. 3, pp. 1491–1495, 2017.
[3] O. E. Taylor and P. S. Ezekiel, "A model to detect spam email using support vector classifier and random forest classifier," *Int. J. Comput. Sci. Math. Theory*, vol. 6, no. 1, pp. 1–11, 2020.
[4] V. Arya, A. A. D. Almomani, A. Mishra, D. Peraković, and M. K. Rafsanjani, "Email spam detection using Naive Bayes and random forest classifiers," in *Proc. Int. Conf. Cyber Secur., Privacy Netw. (ICSPN)*, N. Nedjah, G. Martínez Pérez, and B. B. Gupta, Eds., Cham, Switzerland: Springer, 2023, pp. 341–348.
[5] M. Salb, L. Jovanovic, M. Zivkovic, E. Tuba, A. Elsadai, and N. Bacanin, "Training logistic regression model by enhanced moth flame optimizer for spam email classification," in *Computer Networks and Inventive Communication Technologies*, S. Smys, P. Lafata, R. Palanisamy, and K. A. Kamel, Eds., Singapore: Springer, 2023, pp. 753–768.
[6] N. H. Marza, M. E. Manaa, and H. A. Lafta, "Classification of spam emails using deep learning," in *Proc. 1st Babylon Int. Conf. Inf. Technol. Sci. (BICITS)*, Apr. 2021, pp. 63–68.
[7] B. Kim, S. Abuadbba, and H. Kim, "DeepCapture: Image spam detection using deep learning and data augmentation," in *Information Security and Privacy*, J. K. Liu and H. Cui, Eds., Cham, Switzerland: Springer, 2020, pp. 461–475.
[8] M. Nicho, F. Majdani, and C. D. McDermott, "Replacing human input in spam email detection using deep learning," in *Artificial Intelligence in HCI*, H. Degen and S. Ntoa, Eds., Cham, Switzerland: Springer, 2022, pp. 387–404.
[9] I. AbdulNabi and Q. Yaseen, "Spam email detection using deep learning techniques," *Proc. Comput. Sci.*, vol. 184, pp. 853–858, Jan. 2021.
[10] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection," *Proc. Comput. Sci.*, vol. 190, pp. 479–486, Jan. 2021.
[11] S. Nandhini and J. Marseline K. S., "Performance evaluation of machine learning algorithms for email spam detection," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (IC-ETITE)*, Feb. 2020, pp. 1–4.
[12] N. Kumar, S. Sonowal, and Nishant, "Email spam detection using machine learning algorithms," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 108–113.
[13] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5019–5081, Oct. 2020.
[14] S. Gibson, B. Issac, L. Zhang, and S. M. Jacob, "Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms," *IEEE Access*, vol. 8, pp. 187914–187932, 2020.
[15] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106229.
[16] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive Bayes algorithm for spam filtering," in *Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2016, pp. 1–8.
[17] T. O. Omotehinwa and D. O. Oyewola, "Hyperparameter optimization of ensemble models for spam email detection," *Appl. Sci.*, vol. 13, no. 3, p. 1971, Feb. 2023.
[18] K. Debnath and N. Kar, "Email spam detection using deep learning approach," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COM-IT-CON)*, vol. 1, May 2022, pp. 37–41.
[19] C. M. Shaik, N. M. Penumaka, S. K. Abbireddy, V. Kumar, and S. S. Aravinth, "Bi-LSTM and conventional classifiers for email spam filtering," in *Proc. 3rd Int. Conf. Artif. Intell. Smart Energy (ICAIS)*, Feb. 2023, pp. 1350–1355.
[20] A. Sheneamer, "Comparison of deep and traditional learning methods for email spam filtering," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 560–565, 2021.
[21] S. Kaddoura, O. Alfandi, and N. Dahmani, "A spam email detection mechanism for English language text emails using deep learning approach," in *Proc. IEEE 29th Int. Conf. Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Sep. 2020, pp. 193–198.
[22] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Next-generation spam filtering: Comparative fine-tuning of LLMs, NLPs, and CNN models for email spam classification," *Electronics*, vol. 13, no. 11, p. 2034, May 2024.
[23] H. Yang, Q. Liu, S. Zhou, and Y. Luo, "A spam filtering method based on multi-modal fusion," *Appl. Sci.*, vol. 9, no. 6, p. 1152, Mar. 2019.
[24] S. Zavrak and S. Yilmaz, "Email spam detection using hierarchical attention hybrid deep learning method," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120977.
[25] Z. Chen, R. Tao, X. Wu, Z. Wei, and X. Luo, "Active learning for spam email classification," in *Proc. 2nd Int. Conf. Algorithms, Comput. Artif. Intell.*, New York, NY, USA, Dec. 2019, pp. 457–461.
[26] S. A. Khan, K. Iqbal, N. Mohammad, R. Akbar, S. S. A. Ali, and A. A. Siddiqui, "A novel fuzzy-logic-based multi-criteria metric for performance evaluation of spam email detection algorithms," *Appl. Sci.*, vol. 12, no. 14, p. 7043, Jul. 2022.
[27] G. Nasreen, M. M. Khan, M. Younus, B. Zafar, and M. K. Hanif, "Email spam detection by deep learning models using novel feature selection technique and BERT," *Egyptian Informat. J.*, vol. 26, Jun. 2024, Art. no. 100473.
[28] M. N. Abdal, M. H. K. Oshie, M. A. Haque, and S. Rahman, "A robust model for effective spam detection based on Albert," in *Proc. 6th Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2023, pp. 1–6.
[29] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "A review of spam email detection: Analysis of spammer strategies and the dataset shift problem," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1145–1173, Feb. 2023.
[30] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–17.

JIN-SEONG KIM received the B.E. degree from Yonsei University, Wonju, South Korea, in 2023, where he is currently pursuing the M.S. degree in computer science. His research interests include security for artificial intelligence, natural language processing, and large language models.

HAN-JIN LEE received the B.E. degree from Yonsei University, Wonju, South Korea, in 2024, where he is currently pursuing the M.S. degree in computer science. His research interests include security for artificial intelligence, computer vision, and data generation.

HAN-JU LEE received the B.E. degree from Yonsei University, Wonju, South Korea, in 2024, where he is currently pursuing the M.S. degree in computer science. His research interests include security for artificial intelligence, adversarial examples, and computer vision.

SEOK-HWAN CHOI received the B.E. and Ph.D. degrees from Pusan National University, Busan, Republic of Korea, in 2016 and 2022, respectively. He is currently an Assistant Professor with the Division of Software, Yonsei University, Wonju, Republic of Korea. His research interests include artificial intelligence for security, security for artificial intelligence, adversarial examples, backdoor attacks on deep learning models, malware detection, and intrusion detection systems.