Sumit Ghimire , 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Spam Detection from Organizational Emails Using Machine Learning

Sumit Ghimire[1], Bhoj Raj Ghimire[2]
[1]Research Scholar, Faculty of Science, Health and Technology, Nepal Open University, Lalitpur, Nepal
[2]Faculty Member, Faculty of Science, Health and Technology, Nepal Open University, Lalitpur, Nepal

Abstract- Email communication plays a critical role in modern organizations. The increasing volume of spam, phishing, and malicious emails poses significant security and productivity challenges. This study investigates the effectiveness of machine learning techniques for organizational email spam detection using machine learning-based classification approaches. Enron Email Dataset and a labeled spam collection dataset containing 5,572 messages categorized as ham and spam. The data were preprocessed through text normalization, tokenization, stop-word removal, and feature extraction using sequence encoding and padding techniques. Four deep learning architectures Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Temporal Convolutional Network (TCN) were implemented and optimized. Model performance was evaluated using accuracy, precision, recall, F1-score, ROC-AUC, training time, CPU utilization, and memory consumption. Experimental results showed that all models achieved high classification performance, with accuracy exceeding 97%. The findings indicate that deep learning approaches provide effective and practical solutions for organizational email spam detection by balancing classification accuracy and computational efficiency.

Keywords: Spam Detection, Organizational Email Security, Machine Learning, CNN, LSTM, RNN, TCN, Email Classification.

I. INTRODUCTION

With the rapid growth of technology and internet, emails have become one of the most widely used communication. However, the increasing use of emails has also led to a significant rise in spam messages, such as advertisements, phishing links, or malicious content. These unwanted messages not only disturb users but also create potential security risks such as fraud, identity theft, and malware distribution. Therefore, developing effective techniques for detecting and filtering spam messages has become an important research problem.

Traditional spam detection systems often rely on rule-based filtering or classical machine learning techniques, which may struggle to capture complex patterns in messages. In recent years, deep learning approaches have gained attention due to their ability to automatically learn meaningful features from textual data. Models such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Temporal Convolutional Networks (TCN) have shown great performance in various natural language processing tasks.

In this study, several deep learning models are implemented and evaluated for spam detection. The models are trained on a labeled dataset and their performance is compared using multiple evaluation metrics. In addition to classification accuracy, computational resource consumption is also analyzed to understand the efficiency of each model. The objective of this research is to identify a model that provides both high detection accuracy and efficient resource utilization for practical spam detection systems.

Objectives

- **The objectives are as follows:**
- To review existing literature on detection of spam in organizational email using machine learning techniques.
- To identify and classify legitimate (ham) and spam emails in organizational email communications using machine learning-based classification techniques.
- To compare the performance of deep learning models using evaluation metrics such as

accuracy, precision, recall, F1-score, and ROC–AUC, while analyzing the trade-offs between classification effectiveness, computational complexity, and suitability for real-world organizational deployment.

Problem Domain

Email is one of communication channels used by organizations for internal and external communication. As the volume of email traffic continues to grow, organizations face increasing challenges from spam, phishing, and other malicious email messages. These unwanted emails not only reduce employee productivity but also create serious cybersecurity risks, including data breaches, financial fraud, malware infections, and unauthorized access to sensitive organizational information.

Cybercriminals continuously modify the structure and content of spam emails to bypass filters. Furthermore, modern spam messages often contain sophisticated language, embedded links, and social engineering techniques that are difficult to detect. Machine learning provide solution to this problem by automatically learning patterns from large collections of email data and identifying characteristics that distinguish legitimate (ham) emails from spam messages.

Despite significant progress in spam detection research, organizations still require solutions that not only achieve high classification accuracy but also maintain computational efficiency for practical deployment. Therefore, there is a need to investigate and compare different deep learning architectures to determine their effectiveness in detecting spam emails while considering resource utilization, processing time, and real-world applicability in organizational environments.

II. LITERATURE REVIEW

Managers and other coworkers observed that the offenders had exhibited signs of stress, disgruntlement, or other issues, but no alarms were raised (Greitzer, Kangas, Noonan, & Dalton, 2010). Barriers to using such psychosocial indicators include the inability to recognize the signs and the failure to record the behaviors so that they could be assessed by a person experienced in psychosocial evaluations. The policy of monitoring and surveillance is likely to have a negative effect on employee satisfaction and employee privacy though; it is essential to identify and prevent unacceptable behavior of employees if any (Jahagirdar & Bankar, 2020). Instead of directly monitoring the employee's activity, we could also look out emails with machine learning ability to audit the work.

Enron Corpus emails assist in email-based research, providing a real-world dataset for classification, spam detection, and insider analysis (Klimt & Yang, The Enron Corpus: A New Dataset for Email Classification Research, 2004). Subsequent works helped further, presenting designs, graph-based insights, and statistical analyses of the corpus (Klimt, Introducing the Enron Corpus, 2004). Early spam filtering methods primarily used probabilistic and statistical techniques such as Naïve Bayes (Sahami, Dumais, Heckerman, & Horvitz, 1998) and Support Vector Machines (SVMs) (Drucker, Wu, & Vapnik, 1999) for machine learning applications.

Numerous research in phishing and spam detection is performed. Studies such as PhishRank explored large-scale classification, while Alazab and Choo (Alazab & Choo, 2020) and Khonji et al. (Khonji, Iraqi, & Jones, 2013) were performed on feature engineering and detection methodologies. Deep learning approaches including convolutional, recurrent, and hybrid neural architectures were examined by Vinayakumar (Vinayakumar, Soman, & Poornachandran, 2019), Nandhini and Malarvizhi (Nandhini & Malarvizhi, 2022), and Sadeghzadeh (Sadeghzadeh, 2021) .

Research on insider threat detection has focused on employee's behavioral action. Surveys by Salem (Salem, Hershkop, & Stolfo, 2008), Shabtai and Elovici (Shabtai & Elovici, 2012), and Rezaei and Liu (Rezaei & Liu, 2020) classify threat types. Recent efforts employ deep learning and federated techniques for privacy-preserving, distributed detection.

Various researches that use AI, stats, language to detect anomalous or malicious communication. Studies by Islam and Murase (Islam & Murase, 2022) and Alqahtani and Tawalbeh (Alqahtani & Tawalbeh, 2023) explore semi-supervised and transformer-based techniques, while Chandola (Chandola, Banerjee, & Kumar, 2009) provide theory in anomaly detection that can be used. These works show the works from early spam filtering by using probability to advanced systems for intelligent and secure email analysis.

Early work uses statistical, content-based filtering for spam detection which needs careful feature selection, preprocessing. Androutsopoulos (Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000) evaluated Naïve Bayes and set methodological expectations (feature sets, training size effects, evaluating false-positive costs) that are used nowadays in comparative studies. This early baseline now shifts toward complex representations and more complex model families.

Over the last decade spam detection shifts towards TF–IDF and lexical/url features, word embeddings (Word2Vec), contextual embeddings and fine-tuned transformers (BERT / RoBERTa / DistilBERT), graph/behavior features (sender networks), and anomaly-based signals. Modern researches emphasize combining these into pipelines that balance accuracy, latency, robustness, and explainability.

Alnajjar and Ouni (Alnajjar & Ouni, 2024) performed comparative study evaluates classical ML classifiers (logistic regression, SVM, tree ensembles) across three representation pipelines (TF–IDF, Word2Vec embeddings, and transformer/BERT embeddings). It finds that transformer-based embeddings (BERT) consistently outperform TF–IDF and Word2Vec on standard phishing detection benchmarks, producing much higher precision/recall/F1 in the reported experiments with larger computational cost. These results support the shift toward contextualized language representations for phishing classification. Altwaijry (Altwaijry, Al-Turaiki, Alotaibi, & Alakeel, 2024) benchmarks multiple deep architectures (1D-CNN variants, Bi-GRU hybrids) and reports state-of-the-art performance on several phishing datasets. Their objective was to recommend high-performing architectures and practical hyperparameter choices for deployment.

Alqahtani & Tawalbeh (Alqahtani & Tawalbeh, 2023) evaluates explicitly measure detection accuracy, inference cost, and fine-tuning sensitivity. Such studies highlight tradeoffs where transformers bring large gains in contextual understanding (e.g., detecting obfuscated or contextually malicious wording). Recent transformer-based phishing detection work also explores XAI methods to produce analyst-friendly explanations (Alqahtani & Tawalbeh, 2023).

Semi-supervised / domain adaptation approaches which is described by Islam & Murase (Islam & Murase, 2022) with objectives of reducing labeled data needs while maintaining detection performance under domain shift. Production SSL patterns (student-teacher, self-training) and graph/consistency regularization are frequently recommended to exploit abundant unlabeled email. Surveys on anomaly detection looks out for taxonomy and theoretical grounding for approaches that detect emails or behaviors that significantly deviate from normal patterns. Subsequent email/insider-threat research fuses content with network/graph features (e.g., communication graphs, temporal patterns) to detect spear-phishing and insider threats that content classifiers alone may miss. These works set objectives around early detection of unusual communication patterns rather than only binary phishing classification (Chandola, Banerjee, & Kumar, 2009).

Early work in email filtering established careful empirical methodology and use simple probability that that are used to detect spam emails. Androutsopoulos (Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000) demonstrated how preprocessing choices (tokenization, lemmatization, stop-lists), attribute-set size, training-size, and cost-sensitive evaluation materially influence Naïve Bayes performance; their work therefore became a standard reference for experimental design and baseline reporting in later

studies. Methodological lessons from these early studies e.g., explicitly reporting preprocessing steps and using cost-aware metrics remain even in today's evaluation (Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000).

A major methodological development compares representation pipelines (lexical TF–IDF, static embeddings like Word2Vec, and contextual transformers) because representation choice often dominates performance. Recent comparative studies explicitly evaluate these pipelines end-to-end: Tawil et al. (2024) (Computers, Materials & Continua) compare TF–IDF, Word2Vec, and BERT-based representations across multiple classifiers and find systematic gains for contextual transformer embeddings, while also emphasizing the resource and inference-cost trade-offs associated with fine-tuning large LMs. Such comparative articles typically standardize datasets, vary only the representation and classifier, and report multiple metrics (precision, recall, F1, AUC) so comparison can assess tradeoffs between accuracy and operational cost.

Another contrast studies between supervised learning alforithms (Naïve Bayes, SVM, logistic regression, tree ensembles) with deep learning architectures (CNNs, RNNs, hybrid CNN–RNN) and examines when each family is appropriate (Altwaijry, Al-Turaiki, Alotaibi, & Alakeel, 2024). Methodologically, such papers emphasize hyperparameter sweeps, cross-validation on multiple corpora, and studies to isolate architecture effects.

Transformer-based methodologies introduce additional methodological considerations: tokenization and long-document handling, fine-tuning vs feature-extraction, and explainability hooks. Transformer comparative evaluations (Alqahtani & Tawalbeh, 2023) typically test (a) frozen-embedding + classifier pipelines vs (b) full fine-tuning, and report sensitivity to truncation policies (sliding windows, hierarchical pooling) because many emails exceed standard model token limits. Recent explainable transformer studies also add post-hoc attribution (attention visualizations, LIME/SHAP) to make model outputs actionable for analysts. Methodologically, transformer studies therefore must report tokenization, sequence-length policy, learning-rate schedules, and compute budget to be reproducible.

Many articles emphasize multi-signal fusion (content + metadata + graph/behavioral features) as a methodological best practice, especially for spear-phishing and insider-threat detection where textual cues alone can be insufficient. Methodologies that fuse signals construct feature pipelines that combine: header and sender reputation features, URL lexical features and URL-host reputation, HTML/attachment meta-features, and content embeddings; some studies append graph features (sender–recipient centrality, temporal interaction statistics) and then use either a single multimodal model or stacked ensemble. The anomaly-detection survey by Chandola provides the theoretical taxonomy (statistical, density-based, model-based anomaly detectors) often used when designing behavioral or graph-based detection modules to complement content classifiers (Chandola, Banerjee, & Kumar, 2009).

Training is a second methodological dimension: supervised learning dominates when labeled data are available, but semi-supervised, self-training, and domain-adaptation strategies are increasingly used for enterprise settings with label scarcity or domain shift. Semi-supervised methodological articles demonstrate typical practices such as pseudo-label thresholds, consistency regularization, and careful validation of noisy pseudo-labels; experimental protocols should therefore report unlabeled data sources, pseudo-label selection criteria, and sensitivity analyses to pseudo-label noise. Where privacy is a concern, federated learning or representation-sharing protocols appear as method options, requiring experiments that measure client heterogeneity and communication/computation tradeoffs.

Artificial Neural Networks (ANN) is an arithmetic model that is inspired by how biological neural networks operate. A neural network consists of numerous artificial neurons networked together. They process data by using a connectionist method

of computation. A fundamental rule of ANN is how it modifies its architecture. This depends on both internal and external information ascending through the network while learning. Currently neural networks are non-linear numerical systems. They are frequently deployed to model complex relationships between inputs and outputs to ultimately find patterns in data. Supervised or unsupervised learning can be used on ANN. The difference between unsupervised and supervised training is that there are no expected outputs for the unsupervised learning as opposed to supervised training. The purpose of training is to create an accurate model that predicts the results correctly most of the time. Artificial Neural Network model was implemented for email classification using a feed forward back propagation algorithm for training. The model scored an accuracy of 85.31% on test data which is not bad although they did not specify the computational cost in execution and time (Ssebulime, 2022).

Support Vector Mechanism is a supervised machine learning model that uses classification algorithms in two group classifications problems. As compared with the recent algorithms like Artificial Neural Networks, they have two major advantages like better performance under limited samples and higher speeds. Furthermore, this makes SVM better suited to email classification where it is typical to assess a dataset with numerous tagged samples. With the use of active learning algorithm, the spam filter can achieve a faster level of generalization accuracy in relation to the number of labelled examples. It is also suggested that the online learning algorithm is much faster and saves storage cost because only subsets of the dataset are always considered although the results are combined (Ssebulime, 2022).

Spam emails are unsolicited, often bulk, electronic messages sent for commercial, malicious, or fraudulent purposes. They can contain deceptive links, malware attachments, or phishing attempts. Spam messages not only waste bandwidth and storage but also pose privacy and cybersecurity threats to individuals and organizations (Sahami, Dumais, Heckerman, & Horvitz, 1998).

Spam detection refers to the process of identifying and filtering spam or malicious emails from legitimate communication using machine learning, rule-based, or hybrid techniques. It involves analyzing the content, header, sender reputation, and embedded links to classify messages (Drucker, Wu, & Vapnik, 1999). Modern approaches incorporate deep learning and ensemble methods to enhance detection accuracy (Egele, Kruegel, Vigna, & Kirda, 2009).

Phishing is a deceptive cyberattack technique that tricks users into revealing sensitive information such as login credentials or financial data by impersonating trustworthy entities through email or websites. Phishing often overlaps with spam but is more targeted and harmful (Hassanpour, Dogdu, Choupani, Goker, & Nazli, 2018).

Classification techniques are supervised learning methods that categorize data into predefined classes, such as “spam” vs. “ham” (legitimate). Algorithms like Naïve Bayes, Decision Trees, Random Forests, Support Vector Machines (SVM), and Deep Neural Networks are commonly used in email classification (Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000).

Feature extraction is the process of transforming raw data (like text) into meaningful numerical or categorical features that can be used by machine learning algorithms. In email analysis, features include word frequency, sender metadata, URL patterns, or embeddings derived from natural language processing (NLP) models (Vinayakumar, Soman, & Poornachandran, 2019).

Data cleaning or preprocessing ensures that datasets are consistent, complete, and free of noise or redundancy. In textual datasets like the Enron Corpus, this includes tokenization, removal of stopwords, stemming, lemmatization, and normalization. Proper data cleaning enhances the performance of classification and detection systems (Klimt & Yang, The Enron Corpus: A New Dataset for Email Classification Research, 2004).

Machine Learning is a subset of artificial intelligence that enables systems to learn patterns and make predictions from data without explicit programming. ML underpins most modern email filtering and insider threat detection systems (Chandola, Banerjee, & Kumar, 2009).

Deep Learning is an advanced subset of ML that uses artificial neural networks with multiple layers to automatically learn complex representations from data. DL models such as CNNs, RNNs, LSTMs, and Transformers are widely applied to spam and phishing email detection due to their superior ability to capture contextual semantics (Nandhini & Malarvizhi, 2022).

Accuracy measures the proportion of correctly classified samples among the total instances. However, in imbalanced datasets like spam filtering, accuracy alone can be misleading; therefore, precision, recall, and F1-score are often used together for evaluation (Chandola, Banerjee, & Kumar, 2009).

Precision measures the percentage of correctly identified positive samples among all predicted positives, while recall measures the percentage of actual positives that were correctly identified. Both metrics are critical inevaluating email classifiers, where false positives (legitimate emails marked as spam) must be minimized (Powers, 2011).

Robustness refers to the ability of a system to maintain high performance under noisy, variable, or adverse conditions. (Sabottke & Suciu, 2021). A false positive occurs when a legitimate email is incorrectly labeled as spam, while a false negative happens when a spam email is incorrectly classified as legitimate. Minimizing these errors is crucial for maintaining user trust and system reliability (Drucker, Wu, & Vapnik, 1999).

Insider threats occur when individuals within an organization misuse their authorized access to harm systems, leak data, or compromise integrity. These threats may be malicious (intentional sabotage) or unintentional (negligence) and are often detected through behavioral, content, and graph-based analysis (Salem, Hershkop, & Stolfo, 2008).

Email monitoring involves analyzing email metadata and content to detect security incidents or policy violations. It supports compliance, risk assessment, and insider threat management by observing communication behaviors over time (Le & Markopoulou, 2021).

Audit architectures are systematic frameworks that log, store, and analyze system activities to ensure accountability and detect anomalies. In cybersecurity, audit frameworks facilitate forensic analysis and cross-organizational threat detection (Eberle & Holder, 2009).

III. METHODOLOGY

This study proposes a comparative framework for email spam classification Deep Learning (DL) techniques. The methodology consists of several stages including dataset collection, text preprocessing, feature extraction, model development, hyperparameter optimization, resource monitoring, and performance evaluation. A total of four deep learning models namely RNN, LSTM, CNN, and TCN were implemented and compared.

Dataset Description

Emails were collected from organizational email datasets of Enron Email Dataset which is publicly available. The Enron email dataset contains emails generated by employees of the Enron Corporation. It was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse. For the purpose of this research, the May 7, 2015 version of the dataset, as published by Carnegie Mellon University, was utilized.

The experiments were conducted using the Spam Collection dataset, which contains labeled messages categorized as ham (legitimate messages) and spam (unsolicited messages). The dataset was obtained from a publicly available repository and loaded using attributes namely as Label and Text. Label indicates whether the message is spam or ham whereas text is

the actual email message content for computational efficiency, 5572 samples were used in this study. The dataset distribution indicates a natural imbalance between the classes, with 4825 ham messages and 747 spam messages.

Text Preprocessing: Before training the models, the raw emails were preprocessed to remove noise and improve model performance. The preprocessing pipeline included the following steps:

Lowercase Conversion: All text messages were converted to lowercase to maintain uniformity and avoid duplication of words with different cases.
URL Removal: Web links and URLs were removed using regular expression patterns.

Removal of Non-Alphabetic Characters:
Numbers, punctuation marks, and special symbols were removed to retain only alphabetic characters.
Tokenization: The cleaned text was split into individual words (tokens).

Stopword Removal: Common English stopwords such as the, is, at, and which were removed using the Natural Language Toolkit (NLTK). After preprocessing, the cleaned text was stored as a new feature used for training the classification models.

Label Encoding: The categorical labels (ham and the Pandas library. The dataset contains two attributes namely as Label and Text. Label indicates whether the message is spam or ham whereas text is the actual email message content for computational efficiency, 5572 samples were used in this study. The dataset distribution indicates a natural imbalance between the classes, with 4825 ham messages and 747 spam messages.

Text Preprocessing: Before training the models, the raw emails were preprocessed to remove noise and improve model performance. The preprocessing pipeline included the following steps:

Lowercase Conversion: All text messages were converted to lowercase to maintain uniformity and avoid duplication of words with different cases.
URL Removal: Web links and URLs were removed using regular expression patterns.

Removal of Non-Alphabetic Characters:
Numbers, punctuation marks, and special symbols were removed to retain only alphabetic characters.
Tokenization: The cleaned text was split into individual words (tokens).

Stopword Removal: Common English stopwords such as the, is, at, and which were removed using the Natural Language Toolkit (NLTK). After preprocessing, the cleaned text was stored as a new feature used for training the classification models.

Label Encoding: The categorical labels (ham and spam) were converted into numerical values as 0 and 1 respectively using Label Encoding. This numerical representation is required for deep learning models.
Dataset Splitting: The dataset was divided into training and testing subsets. The split was performed using an 80:20 ratio, where 80% of the data was used for training and 20% for testing.

Feature Extraction: Deep learning models require sequential numerical input. Therefore, text messages were converted into sequences of integers using the Tokenizer utility from TensorFlow/Keras. Each word was mapped to a unique integer index based on its frequency in the corpus.

To ensure consistent input length, sequences were padded using the function with a maximum sequence length of 100 tokens and a vocabulary size limited to 10,000 words.

Deep Learning Models
Deep learning models are capable of automatically learning hierarchical representations of textual data, making them suitable for sequence-based natural language processing tasks. The models evaluated in this study include Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), Convolutional Neural Networks (CNN), and Temporal Convolutional Networks (TCN). Each architecture was designed to capture different characteristics of textual sequences and was

optimized using hyperparameter tuning to achieve improved classification performance.

Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN), which is designed to process sequential data by maintaining hidden states that capture dependencies between words in a sequence. This architecture is particularly useful for text classification because it preserves contextual information from previous tokens while processing the input. The RNN model used in this study consists of an embedding layer that converts input tokens into dense vector representations, followed by a Simple RNN layer responsible for capturing sequential patterns within the email messages. A dropout layer was included to reduce overfitting by randomly deactivating a portion of neurons during training. Finally, a dense output layer with a sigmoid activation function was used to perform binary classification between spam and ham messages. Key hyperparameters, including embedding dimension, number of RNN units, dropout rate, and learning rate, were optimized using the Keras Tuner Random Search approach. Hyperparameters such as embedding dimension, number of RNN units, dropout rate, and learning rate were optimized using Keras Tuner Random Search.

Long Short-Term Memory (LSTM)

The LSTM network extends the traditional RNN. LSTM is an advanced form of RNN that addresses the vanishing gradient problem commonly encountered in standard recurrent networks. It achieves this by incorporating memory cells and gating mechanisms that regulate the flow of information through the network. These mechanisms enable the model to capture long-term dependencies in text sequences more effectively. In this study, the LSTM architecture includes an embedding layer to transform words into vector representations, followed by an LSTM layer responsible for learning sequential dependencies within the text. The output of the LSTM layer is passed to a sigmoid-activated dense layer for binary classification. Additional hyperparameters, including the number of LSTM units, dropout rate, recurrent dropout, and learning rate, were tuned to optimize model performance.

Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) model was also implemented to capture local textual patterns within email messages. CNNs are effective in natural language processing tasks because convolutional filters can detect important n-gram features such as phrases or word combinations that frequently appear in spam messages. The architecture begins with an embedding layer that converts tokens into dense vectors, followed by a one-dimensional convolutional layer (Conv1D) that extracts local features from the input sequences. A global max pooling layer was then applied to reduce the dimensionality of the feature maps and retain the most important features. To prevent overfitting, a dropout layer was included before the final dense layer with sigmoid activation. The CNN model was optimized by tuning parameters such as the number of convolutional filters, kernel size, embedding dimension, dropout rate, and learning rate.

Temporal Convolutional Network (TCN)

Temporal Convolutional Network (TCN) was implemented for sequence modeling. Unlike traditional recurrent models, TCNs rely entirely on convolutional operations while maintaining the ability to capture long-range dependencies in sequential data. This is achieved through the use of dilated convolutions, which expand the receptive field of the network without significantly increasing computational complexity. The TCN architecture used in this study consists of an embedding layer that transforms input text into vector form, followed by a TCN layer that processes the sequences using dilated convolutional filters. The output of the TCN layer is then passed to a dense layer with sigmoid activation to perform binary classification.

Hyperparameter Optimization

Hyperparameter optimization was performed to improve the performance of the deep learning models by identifying the most suitable configuration of model parameters. In this study, the Keras Tuner Random Search method was used to automatically explore multiple combinations of hyperparameters for each deep learning architecture. Random Search was selected because it efficiently samples the search space and often finds

optimal or near-optimal parameter configurations with fewer trials compared to exhaustive search methods.

Several important hyperparameters were considered during the tuning process. These include the embedding dimension, which determines the size of the vector representation for each word; the number of hidden units in recurrent layers such as RNN and LSTM; and the number of convolutional filters used in CNN and TCN models to extract meaningful features from the input sequences. Additional parameters such as kernel size, which controls the receptive field of convolutional layers, and dropout rate, which helps prevent overfitting by randomly disabling neurons during training, were also optimized. Furthermore, the learning rate of the Adam optimizer was tuned to ensure stable and efficient convergence during model training.

For each deep learning model, the tuner evaluated several candidate configurations by training the models on the training dataset and monitoring validation accuracy as the optimization objective. The Random Search process allowed the system to identify the best-performing hyperparameter combinations for each architecture.

During the hyperparameter optimization process, each model trained for five epochs with a batch size of 64, allowing the tuner to evaluate performance efficiently while maintaining reasonable computational cost. The validation dataset was used to monitor model performance during training and guide the selection of optimal hyperparameters.

Resource Monitoring

In addition to predictive performance, this study also evaluated the computational efficiency of the implemented models by monitoring system resource usage during the training process. Resource monitoring was incorporated to analyze the computational cost associated with different machine learning and deep learning architectures. Since deep learning models often require significantly higher computational resources compared to traditional machine learning methods, measuring system performance provides a more comprehensive comparison between the models.

Psutil library was used to track system resource consumption in real time while each model was being trained. A dedicated monitoring function was implemented to record important metrics including training time, CPU utilization, and memory usage. Training time was measured in seconds to determine how long each model required to complete the training process. Average CPU utilization was recorded as a percentage to reflect the computational workload imposed on the processor during training. In addition, the maximum RAM usage was measured in megabytes (MB) to identify the peak memory consumption of each model.

The monitoring process was executed concurrently with the model training procedure using a background thread that continuously sampled CPU and memory usage at regular intervals. Once the training process was completed, the monitoring function returned the total training time, the average CPU usage across the training period, and the maximum memory usage observed. This resource monitoring approach enabled a detailed comparison of the computational demands of the different models. By combining resource utilization metrics with predictive performance, the study provides a more comprehensive evaluation of each model's practical applicability in real-world spam detection systems.

Performance Evaluation

The performance of the proposed spam classification models was evaluated using several standard classification metrics derived from the confusion matrix. These metrics provide quantitative measures of how effectively each model distinguishes between spam and legitimate messages. The confusion matrix summarizes the prediction outcomes by comparing the predicted labels with the actual labels in the test dataset, allowing for the calculation of multiple evaluation metrics.

The primary evaluation metric used in this study is accuracy, which represents the proportion of

correctly classified messages among all predictions. While accuracy provides an overall indication of model performance, it may not fully capture the effectiveness of spam detection in imbalanced datasets. Therefore, additional metrics including precision, recall, and F1-score were also considered. Precision measures the proportion of messages predicted as spam that are actually spam, which reflects the model's ability to minimize false positives. Recall measures the proportion of actual spam messages that are correctly identified by the model, indicating the effectiveness of the classifier in detecting spam. The F1-score, which is the harmonic mean of precision and recall, provides a balanced evaluation metric when dealing with imbalanced class distributions.

In addition to these metrics, the study also utilized Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) to further analyze model performance. The ROC curve illustrates the relationship between the true positive rate and the false positive rate at different classification thresholds. The AUC value summarizes the overall ability of a model to distinguish between spam and ham messages, with higher values indicating better discriminatory capability. By combining confusion matrix-based metrics with ROC-AUC analysis, a comprehensive evaluation of the classification models was achieved.

IV. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the implementation of deep learning models for email spam classification. The models were evaluated using standard performance metrics including accuracy, precision, recall, and F1-score, along with ROC curve analysis and computational resource monitoring.

The experimental results indicate that all implemented models achieved high classification accuracy, demonstrating the effectiveness of deep learning techniques in identifying spam messages. In addition to classification performance, computational efficiency was also analyzed. The findings highlight the importance of balancing predictive performance with computational efficiency when selecting models for practical spam detection systems.

The dataset consisted of 5572 email messages, including 4825 ham messages and 747 spam messages. After preprocessing and feature transformation, the dataset was divided into training and testing subsets. The performance of the implemented models was evaluated using accuracy, precision, recall, and F1-score. In addition, confusion matrices and computational resource usage were analyzed to provide a comprehensive comparison between the models.

Hyperparameter Tuning Results

The hyperparameter tuning process identified the optimal configurations for each deep learning model.

Table 1 Performance and Computational Resource Comparison of Deep Learning Models for Email Spam Detection

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (sec)	Avg. CPU (%)	Max RAM (MB)
RNN	97.76	93.66	89.26	91.41	28.05	74.51	1222.03
LSTM	98.65	97.18	92.62	94.85	79.09	73.53	1223.85
CNN	99.28	99.30	95.30	97.26	22.04	73.39	1255.81
TCN	98.74	97.87	92.62	95.17	24.04	57.31	1213.01

For the RNN model, the best-performing configuration consisted of an embedding dimension of 128, 128 Simple RNN units, a dropout rate of 0.2, and an Adam optimizer learning rate of 0.01.

For the LSTM model, the optimal hyperparameters included an embedding dimension of 64, 128 LSTM units, no dropout (0.0), no recurrent dropout (0.0), and a learning rate of 0.01.

The CNN model obtained its best performance with an embedding dimension of 128, 32 convolutional filters, a kernel size of 5, a dropout rate of 0.2, and a learning rate of 0.01.

For the TCN model, the optimal configuration consisted of an embedding dimension of 32, 32 filters, a kernel size of 7, a dilation pattern of (1, 2, 4), a dropout rate of 0.2, and a learning rate of 0.01.

Overall, the hyperparameter optimization process significantly improved the predictive performance of all models. The findings indicate that convolution-based architectures were particularly effective for email spam detection, providing high classification accuracy while maintaining relatively low computational cost.

Training Performance of Models

The training performance of the implemented deep learning models was evaluated by analyzing the accuracy and loss values obtained during the training process. Each model was trained using the optimal hyperparameters identified through the hyperparameter tuning phase. The training results indicate that all models successfully learned the underlying patterns within the email spam dataset and achieved high classification performance.

The Recurrent Neural Network (RNN) model demonstrated consistent learning throughout the training process. During the first epoch, the model achieved a training accuracy of approximately 98.36% with a loss value of 0.0579. As training progressed, both accuracy and loss improved substantially. By the fifth epoch, the model reached a training accuracy of 100% with a very low loss value of 0.0004, indicating effective convergence.

The Long Short-Term Memory (LSTM) model exhibited strong training performance from the beginning of the training process. The model achieved an initial training accuracy of approximately 99.51% with a loss value of 0.0197 in the first epoch. The accuracy continued to improve across subsequent epochs while the loss decreased significantly. By the final epoch, the model attained 100% training accuracy with a loss value close to zero (0.0002), demonstrating excellent learning capability. The Convolutional Neural Network (CNN) model also showed outstanding training performance. In the first epoch, the model achieved a training accuracy of approximately 99.46% with a loss value of 0.0285. The model rapidly converged during subsequent epochs, reaching nearly perfect classification performance. At the end of the fifth epoch, the CNN achieved 100% training accuracy with a loss value of 0.0002.

Table 2: Hyperparameters of Models

Model	Best Hyperparameters
RNN	Embedding=128, RNN Units=128, Dropout=0.2, Learning Rate=0.01
LSTM	Embedding=64, LSTM Units=128, Dropout=0.0, Recurrent Dropout=0.0, Learning Rate=0.01
CNN	Embedding=128, Filters=32, Kernel Size=5, Dropout=0.2, Learning Rate=0.01
TCN	Embedding=32, Filters=32, Kernel Size=7, Dilations=(1,2,4), Dropout=0.2, Learning Rate=0.01

The Temporal Convolutional Network (TCN) model achieved high training accuracy during training as well. The model started with a training accuracy of approximately 99.92% and a loss value of 0.0102 in the first epoch. Although trained with early stopping based on validation performance, the model maintained high accuracy and low loss throughout the training process. The validation accuracy remained above 97%, demonstrating strong generalization ability while avoiding overfitting.

Overall, the training results indicate that all deep learning models were capable of effectively learning discriminative features for email spam classification. The continuous increase in training accuracy and corresponding reduction in loss values across epochs demonstrate successful model optimization. Among the evaluated models, CNN, LSTM, and TCN exhibited particularly rapid convergence and achieved near-perfect training performance, while the RNN model also demonstrated strong learning capability with slightly slower convergence.

Classification Performance

The classification performance of the implemented deep learning models was evaluated using standard metrics derived from the confusion matrix, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the models' ability to correctly identify spam and ham emails.

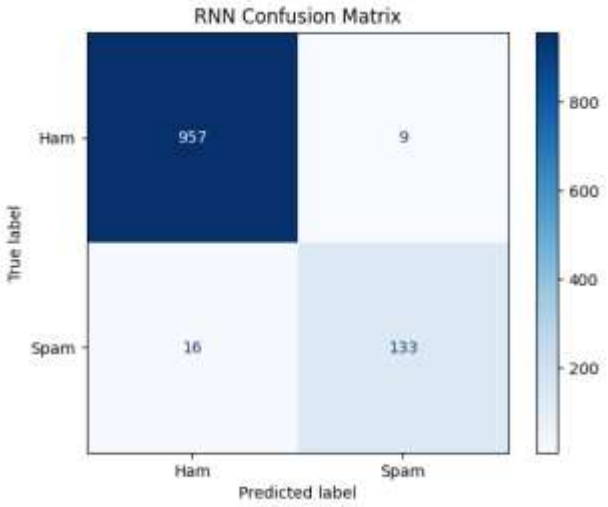

Figure 2: RNN Confusion Matrix

As shown in Figure 2: RNN Confusion Matrix, the Recurrent Neural Network (RNN) model achieved an overall accuracy of 97.76%, indicating strong performance in classifying email messages as ham or spam. The model obtained a precision of 93.66%, showing that most messages predicted as spam were actually spam, while the recall of 89.26% indicates that the model successfully identified the majority of spam messages present in the dataset. The resulting F1-score of 91.41% demonstrates a good balance between precision and recall. The confusion matrix reveals that the model correctly classified 957 ham messages and 133 spam messages, while 9 ham messages were incorrectly classified as spam and 16 spam messages were misclassified as ham. These results suggest that the tuned RNN model effectively distinguished between legitimate and spam email messages, with relatively few classification errors.

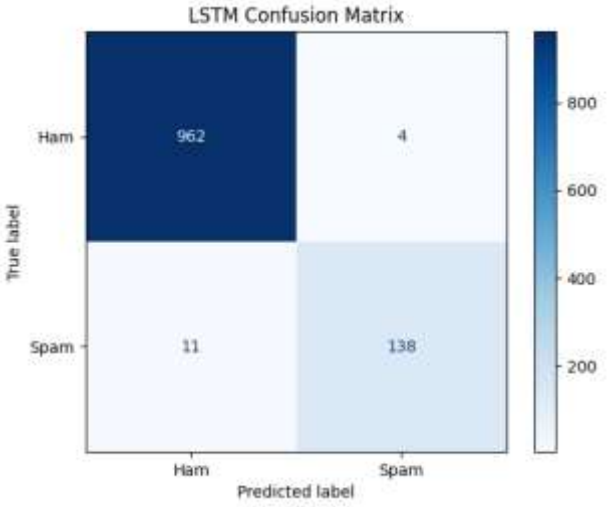

Figure 3: LSTM Confusion Matrix

As shown in Figure 3, the Long Short-Term Memory (LSTM) model achieved an overall accuracy of 98.65%, demonstrating excellent performance in email spam classification. The model obtained a precision of 97.18%, indicating that nearly all messages predicted as spam were correctly identified, while the recall of 92.62% shows that the model successfully detected a large proportion of actual spam messages. The resulting F1-score of 94.85% reflects a strong balance between precision and recall, highlighting the effectiveness of the LSTM architecture in capturing sequential patterns within email text data.

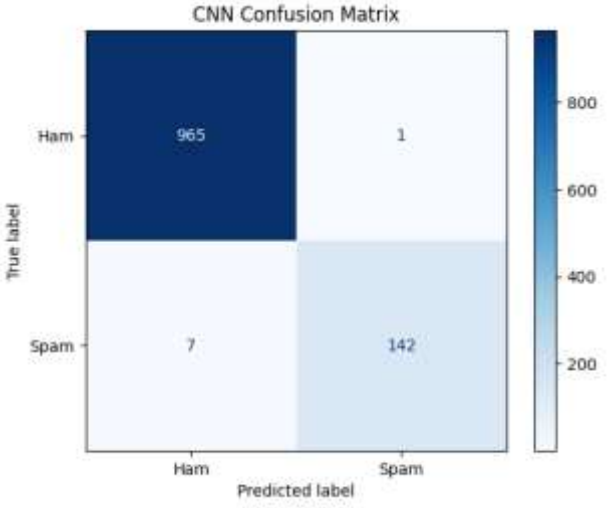

Figure 4: CNN Confusion Matrix

The confusion matrix shows that the model correctly classified 962 ham messages and 138 spam messages, while only 4 ham messages were incorrectly classified as spam and 11 spam messages were misclassified as ham.

As shown in Figure 4, the Convolutional Neural Network (CNN) model achieved an overall accuracy of 99.28%, making it the best-performing model among the evaluated deep learning architectures for email spam detection. The model obtained a precision of 99.30%, indicating that almost every message predicted as spam was correctly classified. The recall of 95.30% demonstrates that the CNN successfully identified the vast majority of actual spam messages in the dataset.

The resulting F1-score of 97.26% reflects an excellent balance between precision and recall, highlighting the model's strong classification capability. The confusion matrix shows that the model correctly classified 965 ham messages and 142 spam messages, while only 1 ham message was incorrectly classified as spam and 7 spam messages were misclassified as ham.

As shown in Figure 5, the Temporal Convolutional Network (TCN) model achieved an overall accuracy of 98.74%, demonstrating excellent performance in email spam classification. The model obtained a precision of 97.87%, indicating that nearly all messages predicted as spam were correctly identified. The recall of 92.62% shows that the model successfully detected a large proportion of actual spam messages in the dataset.

The resulting F1-score of 95.17% reflects a strong balance between precision and recall, highlighting the effectiveness of the TCN architecture in capturing both local and long-range dependencies within email text sequences. The confusion matrix reveals that the model correctly classified 963 ham messages and 138 spam messages, while 3 ham messages were incorrectly classified as spam and 11 spam messages were misclassified as ham.

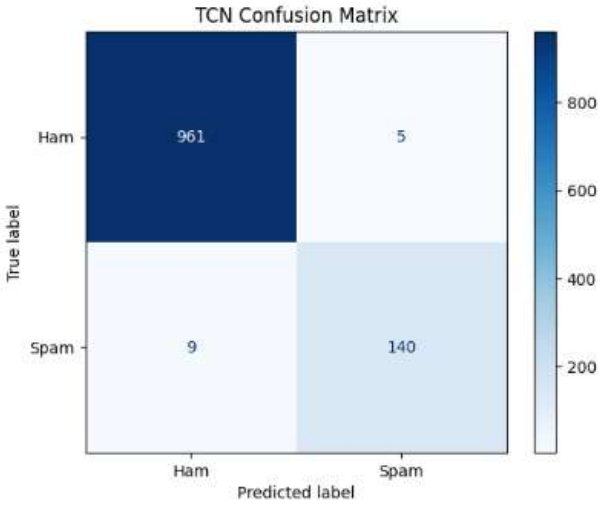

Figure 5: TCN Confusion Matrix

As shown in Figure 6, ROC Curve, Receiver Operating Characteristic (ROC) curve is used to evaluate the classification performance of models by illustrating the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at various classification thresholds. The True Positive Rate, also known as recall, represents the proportion of correctly identified spam messages among all actual spam messages. The False Positive Rate represents the proportion of ham emails incorrectly classified as spam. In the ROC curve, the x-axis represents the False Positive Rate, while the y-axis represents the True Positive Rate.

The dashed diagonal line represents the performance of a random classifier. Any model with a curve above this line performs better than random guessing. The closer the ROC curve is to the top-left corner of the graph, the better the model is at distinguishing between spam and ham emails.

To quantify the ROC performance, the Area Under the Curve (AUC) metric was used. A higher AUC value indicates superior classification capability. From the results, the LSTM model achieved the highest AUC value of 0.993, indicating the strongest discriminative performance among the evaluated models. The TCN model achieved an AUC of 0.989, followed closely by the RNN model with an AUC of 0.982. The CNN model obtained an AUC value of 0.986, which also reflects excellent classification capability. Since all AUC values exceeded 0.98, each

model demonstrated a very high ability to separate spam messages from legitimate email messages.

Overall, the results show that all deep learning models performed highly effectively for email spam classification, achieving accuracy values above 97%. Among the evaluated models, the CNN model achieved the highest accuracy (99.28%) and F1-score (97.26%), making it the most effective model for the classification task in this study. The TCN and LSTM models also demonstrated excellent performance, with strong precision, recall, and AUC values. Although the RNN model achieved slightly lower performance metrics compared to the other architectures, it still provided reliable classification results and maintained a high level of predictive accuracy.

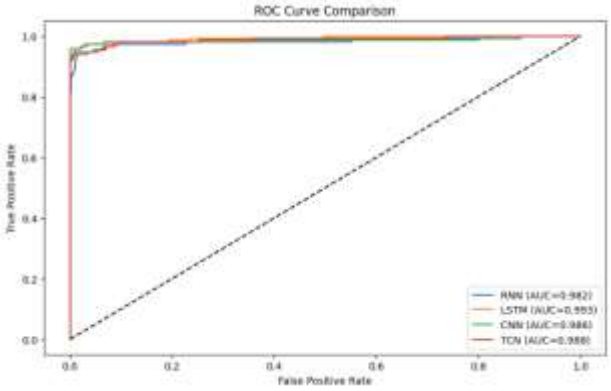

Figure 6 ROC Curve

Performance and Computational Resource Consumption

In addition to classification performance, the computational efficiency of the deep learning models was evaluated using training time, CPU utilization, and memory consumption. These metrics provide insight into the practical feasibility of deploying the models in real-world email spam detection systems where computational resources may be limited.

The results indicate notable differences in resource requirements among the evaluated models. The LSTM model required the highest computational cost, with a training time of 79.09 seconds, an average CPU utilization of 73.53%, and an average memory consumption of 1223 MB. The increased training time can be seen in the LSTM architecture, which improve sequence learning capability but require additional computational overhead.

The RNN model achieved a training time of 28.05 seconds, with an average CPU utilization of 74.51% and an average memory usage of 1222 MB. Although the RNN required considerably less training time than the LSTM model, its classification performance was also comparatively lower.

Among all evaluated models, the CNN model demonstrated the fastest training process, requiring only 22.04 seconds of training time. The model utilized an average CPU percentage of 73.39% and consumed approximately 1255 MB of memory. Despite its low computational cost, the CNN achieved the highest classification accuracy and F1-score, indicating balance between effectiveness and efficiency.

The TCN model required 24.04 seconds of training time, making it the second-fastest model after CNN. Furthermore, it exhibited the lowest average CPU utilization of 57.31% and the lowest average memory consumption of 1213 MB among all models. These results demonstrate the computational efficiency of the TCN architecture while maintaining strong classification performance.

Overall, the CNN and TCN models offered the most favorable trade-off between predictive performance and computational resource consumption. While the LSTM model achieved competitive classification results, its significantly higher training time makes it less computationally efficient. The CNN model emerged as the most practical solution for email spam detection, achieving the highest accuracy (99.28%) with the lowest training time, whereas the TCN model provided the most resource-efficient execution in terms of CPU and memory utilization.

Discussion of Results

The experimental results shows that deep learning architectures are highly effective for email spam detection. All implemented models achieved high classification accuracy and strong precision, recall, and F1-score values. These findings are consistent with previous studies that highlight the effectiveness

of machine learning and deep learning techniques for detecting malicious or suspicious communications in digital environments.

The strong classification performance observed in this study aligns with the work of Benkahla and Aissa (Benkahla & Aissa, 2024), who demonstrated that machine learning models such as gradient boosting ensembles can effectively detect suspicious emails in real-time systems. Similarly, Egele (Egele, Ensemble and Deep Learning Techniques for Phishing and Spam Detection - Comparative Study, 2020) reported that ensemble and deep learning approaches significantly improve spam and phishing detection performance when compared to traditional rule-based methods. The high accuracy achieved by the deep learning models in this study further supports the conclusion that advanced machine learning techniques are well suited for identifying patterns associated with spam or malicious communication.

Among the evaluated models, the CNN model achieved the highest classification accuracy and F1-score. This result supports the findings of Hassanpour (Hassanpour, Choupani, & Goker, Deep Learning Approaches for Phishing Detection: A Comparative Study, 2018), who showed that deep learning architectures, particularly convolutional models, are highly effective for detecting phishing messages by learning textual patterns automatically. CNN models are particularly effective in extracting local contextual features and phrase patterns, which are common characteristics in spam messages such as promotional keywords or repetitive structures.

The performance of the RNN and LSTM models also demonstrates the importance of sequential modeling in text-based threat detection. Previous research by Haq et al. (Haq, Khan, & Alshehri, 2022) emphasized the role of natural language processing and word embedding techniques in detecting insider threats in communication systems. Similarly, Pennada et al. (Pennada, Nayak, & Krishna, 2025) showed that combining behavioral analysis with machine learning and deep learning techniques improves the detection of insider threats in communication networks. The ability of RNN and LSTM models to capture sequential dependencies in text contributes to their strong performance in identifying spam messages.

The TCN model also produced competitive classification results, confirming that convolution-based architectures designed for sequential data can effectively capture long-range dependencies in textual sequences. This observation aligns with the findings of Cao (Cao, 2024), who demonstrated the effectiveness of convolutional neural networks for threat detection in distributed environments using federated learning techniques. The dilated convolution mechanism used in TCN architectures allows the model to analyze both short-term and long-term patterns within message sequences.

In addition to predictive performance, computational efficiency plays an important role in evaluating the practicality of machine learning models. The resource monitoring results in this study revealed significant differences in training time and system resource usage across models.

The importance of analyzing both behavioral and content-based features in communication security has also been emphasized in previous studies. Popescu-Negrea and Anderson (Popescu-Negrea & Anderson, 2019) demonstrated that behavioral modeling techniques can enhance insider threat detection in enterprise email systems, while Hosseini and Nguyen (Hosseini & Nguyen, 2020) showed that combining content and network features improves threat detection accuracy in email datasets such as Enron. Although the present study focuses on spam detection, the results suggest that similar deep learning approaches could be applied to broader communication security problems, including phishing detection and insider threat monitoring.

V. CONCLUSION

This study presented a comparative analysis of deep learning approaches for email spam classification. A methodology was implemented to clean and prepare textual data, followed by feature extraction using TF-IDF for traditional machine learning models and tokenization with sequence padding for deep learning models. Four deep learning architectures

including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Temporal Convolutional Network (TCN), were evaluated.

The process of machine learning to detect spam in emails can be applied in Enterprise email security systems (e.g., detecting phishing, business email compromise, insider threats), Email gateways and SOC (Security Operations Center) platforms, where it automatically flags or quarantines suspicious messages, Privacy-preserving detection frameworks, where sensitive text and structural features may be processed separately for compliance.

The system can also be designed to function in real-time email filtering or forensic analysis of historical mailboxes. Experimental results demonstrated that all models achieved high classification accuracy; however, performance and resource usage varied depending on the architecture. highlighting the trade-off between predictive performance and computational cost.

The findings indicate that deep learning architectures can effectively capture sequential patterns in text data and improve spam detection performance. Future work may explore larger datasets, advanced architectures such as transformer-based models, and techniques for handling class imbalance to further improve spam detection systems.

REFERENCES

1. Alazab, M., & Choo, K.-K. (2020). Detecting Phishing Emails Using Machine Learning: A Survey and Design Recommendations. IEEE Access.
2. Alnajjar, F., & Ouni, A. (2024). A Comparative Study of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT. Computers, Materials & Continua, 3395-3412. doi:10.32604/cmc.2024.057279
3. Alqahtani, H., & Tawalbeh, L. (2023). Transformer-based Phishing Email Detection: A Comparative Evaluation. Journal of Information Security & Applications.
4. Altwaijry, N., Al-Turaiki, I., Alotaibi, R., & Alakeel, F. (2024). Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models. Sensors, 24(7), 2077. doi:10.3390/s24072077
5. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., & Spyropoulos, C. (2000). An Evaluation of Naive Bayesian Anti-Spam Filtering.
6. Benkahla, S., & Aissa, A. (2024). Real-time Suspicious Email Detection Using Gradient Boosting Ensembles. Applied AI Journal.
7. Cao, Y. (2024). Federated Learning for Cross-Organization Insider Threat Detection: Imageized Feature Representations and CNNs. Scientific Reports.
8. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), 1-58. doi:10.1145/1541880.1541882
9. Drucker, H., Wu, D., & Vapnik, V. (1999). Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks.
10. Eberle, W., & Holder, L. (2009). Insider Threat Detection Using Graph and Anomaly Mining. Journal of Digital Forensics, Security and Law.
11. Egele, M. (2020). Ensemble and Deep Learning Techniques for Phishing and Spam Detection - Comparative Study.
12. Egele, M., Kruegel, C., Vigna, G., & Kirda, E. (2009). PhishRank: Large-Scale Phishing Email Detection. ACM Conference on Computer and Communications Security.
13. (2015). Enron Email Dataset. Carnegie Mellon University.
14. Greitzer, F., Kangas, L., Noonan, C., & Dalton, A. (2010). Identifying at-risk employees: A behavioral model for predicting potential insider threats.
15. Haq, M., Khan, M., & Alshehri, M. (2022). Insider Threat Detection Based on NLP Word Embedding and Machine Learning. Intelligent Automation & Soft Computing, 619-635. doi:10.32604/iasc.2022.021430
16. Hassanpour, R., Choupani, R., & Goker, O. (2018). Deep Learning Approaches for Phishing Detection: A Comparative Study. doi:10.1145/3190645.3190719

17. Hassanpour, R., Dogdu, E., Choupani, R., Goker, O., & Nazli, N. (2018). Phishing E-mail Detection by Using Deep Learning Algorithms. doi:10.1145/3190645.3190719
18. Hosseini, M., & Nguyen, T. (2020). Detecting Insider Threats Using Content and Network Features: An Enron Case Study.
19. Islam, S., & Murase, K. (2022). Semi-supervised Learning for Email Risk Classification in Enterprises.
20. Jahagirdar, R., & Bankar, S. (2020). Employees Perception of Workplace Monitoring and Surveillance. PEOPLE: International Journal of Social Sciences, 474-486. doi:10.20319/pijss.2020.61.474486
21. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing Detection: A Literature Survey. IEEE Communications Surveys & Tutorials, 15(4), 2091-2121. doi:10.1109/SURV.2013.042313.00155
22. Klimt, B. (2004). Introducing the Enron Corpus.
23. Klimt, B., & Yang, Y. (2004). The Enron Corpus: A New Dataset for Email Classification Research. doi:10.1007/978-3-540-30115-8_22
24. Le, T., & Markopoulou, A. (2021). Email-based Insider Threat Detection Using Topic Modeling and Supervised Learning.
25. Nandhini, M., & Malarvizhi, K. (2022). Phishing Email Detection Using Hybrid Feature Extraction and Deep Learning.
26. Pennada, S., Nayak, S., & Krishna, V. (2025). Insider Threat Detection Using Behavioural Analysis through Machine Learning and Deep Learning Techniques. International Research Journal of Multidisciplinary Technovation, 74-86. doi:10.54392/irjmt2527
27. Popescu-Negrea, M., & Anderson, R. (2019). Behavioural Modeling for Insider Threat Detection in Enterprise Email.
28. Powers, D. (2011). Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. Journal of Machine Learning Technologies, 37-63.
29. Rezaei, S., & Liu, X. (2020). A Survey on the Detection of Insider Threats: Behavioural, Content and Graph Based Methods.
30. Sabottke, C., & Suciu, O. (2021). Adversarial Attacks on Phishing Detectors and Robustness Measures.
31. Sadeghzadeh, M. (2021). Phishing Detection via Transformer Embeddings and Similarity Search.
32. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. AAAI Workshop on Learning for Text Categorization.
33. Salem, M., Hershkop, S., & Stolfo, S. (2008). A Survey of Insider Threat Taxonomies, Analysis, Modeling and Countermeasures.
34. Shabtai, A., & Elovici, Y. (2012). A Taxonomy and Survey of Insider Threat Research. Journal of Information Assurance & Security.
35. Ssebulime, T. (2022). Email Classification Using Machine Learning Techniques. ResearchGate Preprint, 1-10.
36. Vinayakumar, R., Soman, K., & Poornachandran, P. (2019). Deep Learning Approaches for Spam and Phishing Detection: A Survey.

.