

Paper Type: Original Article

Enhanced Email Spam Detection Using Fine-Tuned Transformer Models and Classical Machine Learning Techniques

Ahmed Abdelhafeez ^{1,*}  and Alber S. Aziz ² 

¹ Faculty of Computer and Information Technology, Innovation University, Cairo, Egypt; ahmed.abdehafeez@iu.edu.eg.

² Faculty of Information, Systems and Computer Science, October 6th University, Cairo, 12585, Egypt; albershawky.csis@o6u.edu.eg.

Received: 31 Jan 2025

Revised: 03 May 2025

Accepted: 17 Sep 2025

Published: 19 Sep 2025

Abstract

Email spam continues to be a pervasive challenge, causing significant disruptions to users and organizations by wasting time, reducing productivity, and posing cybersecurity risks. Traditional spam filtering techniques have achieved reasonable accuracy but often fall short when confronted with increasingly sophisticated spam tactics. Recent advancements in natural language processing, particularly transformer-based models like BERT, offer new opportunities to enhance spam detection by capturing deeper semantic and contextual information within email content. In this article presents a comprehensive comparative study of spam detection methods, combining classical machine learning classifiers with state-of-the-art fine-tuned transformer models from the Hugging Face library. We implement and evaluate standard classifiers alongside a fine-tuned BERT model for spam classification. The results demonstrate that fine-tuned transformer models outperform traditional approaches, achieving higher accuracy, precision, recall, and F1-scores while significantly reducing false positives and negatives. These improvements underline the value of applying transfer learning and deep contextual understanding to address evolving spam strategies. The study discusses the methodology for preprocessing, model training, and evaluation in detail, and includes insights into the trade-offs between model complexity and computational costs. The findings suggest that integrating fine-tuned transformer models into existing spam filtering systems can substantially improve detection robustness and reliability, paving the way for more secure and efficient email communication. Finally, we outline future directions for research, including exploration of larger transformer architectures and ensemble methods to further enhance spam detection performance.

Keywords: Spam Detection; Naive Bayes; Text Classification; Natural Language Processing; Spam Filtering; Fine-Tuned Transformer Models; BERT Model Fine-Tuning.

1 | Introduction

Email has become one of the most widely used communication systems, enabling individuals and organizations to exchange messages quickly and at minimal cost while maintaining reliable delivery. Its popularity stems from its accessibility, ease of use, and the availability of free services, which have made it the dominant medium for online communication [1]. Beyond serving as a tool for composing, sending, receiving, and storing messages, email has evolved into a critical repository of information. However, as reliance on email continues to grow, so do the challenges of managing it effectively. Mismanagement, along with various operational issues, has intensified with the rapid expansion of email usage [2].



Corresponding Author: ahmed.abdehafeez@iu.edu.eg



<https://doi.org/10.61356/j.mawa.2025.8597>



Licensee **Multicriteria Algorithms with Applications**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

The widespread adoption of affordable, high-speed Internet access, along with declining storage costs, has led to a massive surge in email traffic per user. Distinguishing between useful and irrelevant emails has therefore become increasingly complex. Among these challenges, unsolicited email, commonly referred to as spam, junk mail, or bulk email, poses one of the most significant threats to email communication today. Spam is generally defined as unwanted messages sent to large groups of recipients, often without their consent [2, 3]. Such emails typically promote fraudulent schemes, questionable products, or illegal services, ranging from “get rich quick” opportunities and multilevel marketing pitches to pirated software and financial scams. Some spam messages even attempt to sell real estate, pharmaceuticals, or investment offers. Because spam overloads inboxes and consumes network resources, it is recognized as a major threat to Internet efficiency. In fact, the global economic loss attributed to spam-related productivity waste rose by \$2 billion in a single year, reaching \$132 billion in 2010 [4].

Beyond wasting bandwidth and resources, spam also creates a wide range of harmful consequences. Overloaded email servers may fail to process legitimate communication, leading to delayed or lost critical messages. Users are burdened with the time-consuming task of sorting and deleting these messages, while overall Internet performance suffers due to wasted capacity. More critically, spam is a common carrier of malware such as worms, Trojans, and other malicious programs that compromise operating systems and user security. The financial risks are equally concerning, as millions of individuals worldwide fall victim to fraudulent activities propagated through spam each year [5]. These wide-ranging effects have pushed researchers and industry leaders to explore innovative strategies for spam detection, prevention, and eventual elimination.

Spam is available in a wide variety of languages, including Arabic, Korean, Chinese, and other Asian dialects, but is most frequently written in English. As a result, manually classifying too many emails is difficult, and machine learning techniques must be introduced. The machine learning technique entails developing filters to examine and eliminate characteristics from a corpus of spam [6]. Spam detection is not a typical text categorization task since it has some intriguing characteristics. Both spam and legitimate messages can cover a wide variety of topics. Traditional spam filtering techniques started primarily as rule-based systems that relied on predefined keywords, blacklists, and heuristic rules to identify and block unwanted messages. While initially effective, these methods struggle against ever-evolving spam tactics such as obfuscation, use of natural language, and spam generated by increasingly sophisticated language models[7]. As a result, more advanced techniques leveraging machine learning have been developed. Classical machine learning algorithms, including Naive Bayes, Decision Trees, Random Forests, and Support Vector Machines (SVM), learn from labeled examples to classify emails as spam or legitimate (ham), providing higher accuracy and adaptability than manual rule sets.

Recent advances in Natural Language Processing (NLP), specifically the introduction of large pre-trained transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers), have revolutionized text classification tasks, including spam detection. These models capture complex semantic and contextual relationships within text by leveraging attention mechanisms and bidirectional training, enabling them to detect subtle cues in spam emails that traditional methods can miss. Fine-tuning such models on domain-specific spam datasets allows for enhanced performance in real-world scenarios.

Recent advances in Natural Language Processing (NLP), specifically the introduction of large pre-trained transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers), have revolutionized text classification tasks, including spam detection. These models capture complex semantic and contextual relationships within text by leveraging attention mechanisms and bidirectional training, enabling them to detect subtle cues in spam emails that traditional methods can miss. Fine-tuning such models on domain-specific spam datasets allows for enhanced performance in real-world scenarios.

This article will be structured as follows:

- A detailed comparative analysis of classical machine learning classifiers versus fine-tuned transformer models for email spam detection.

- An end-to-end methodology covering dataset preprocessing, model training, and evaluation tailored for robust spam classification.
- Empirical evidence demonstrates that fine-tuning BERT significantly enhances spam detection performance, with notable reductions in false positives and negatives.
- Insights into practical considerations for deploying such hybrid systems in real-time email filtering environments.
- Recommendations for future work, including exploration of larger transformer architectures and ensemble modeling to further improve spam classification.

The rest of the article is organized as follows: Section 2 reviews related work and existing spam detection techniques, Section 3 details the dataset and preprocessing steps, Section 4 describes the methodologies employed, Section 5 presents experimental results, Section 6 discusses findings and practical implications, and finally, Section 7 concludes the article and proposes directions for future research.

2 | Literature Review

Spam email detection has been a vital area of research and development since the inception of email communication. Early spam filtering efforts mainly employed rule-based systems, which relied on heuristic rules like blacklists, whitelists, and keyword pattern matching. These systems analyzed email headers and content to block messages that matched predefined criteria. However, as spammers evolved their tactics by obfuscating text, mimicking valid email formats, and using various evasion techniques, rule-based systems became less effective and required frequent manual updating.

2.1 | Classical Machine Learning Approaches

With the rise of machine learning (ML) techniques, spam detection shifted towards data-driven methods where classifiers learn from large, labeled datasets. Among these, Naive Bayes (NB) emerged as one of the earliest and most widely adopted models due to its simplicity, speed, and surprisingly high accuracy in handling text classification tasks. Naive Bayes operates under the assumption of conditional independence between features (words), which, despite being a simplification, performs well in many practical applications. Studies have shown NB models achieve accuracy in the range of 95% to 98% when applied to email spam datasets.

Decision Trees (DT) are another popular choice, performing classification by recursively partitioning the feature space and making decisions at each node based on feature values. DT models are interpretable and allow for intuitive understanding of decision pathways. However, they tend to overfit on training data unless pruning strategies are applied. Random Forests (RF), an ensemble learning technique that aggregates multiple decision trees trained on different bootstrap samples and feature subsets, improve robustness and accuracy, often yielding performance comparable to or better than NB. However, RF models demand more computational resources and can be less interpretable[8].

Support Vector Machines (SVMs) are also extensively used, leveraging hyperplanes to separate spam from ham in high-dimensional feature spaces. SVMs perform well in handling complex boundaries but require careful tuning of kernel functions and parameters. Table 1 summarizes the typical accuracy reported for these classical models, demonstrating the competitive baseline provided for spam filtering.

2.2 | Advances in Deep Learning for Spam Detection

The emergence of deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), introduced powerful tools for automatic feature extraction from text and capturing sequential dependencies, respectively. CNNs efficiently identify local textual patterns through convolutional filters, which help in recognizing important phrases indicative of spam. RNNs, and their variants such as

Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), model longer-range dependencies and context, improving classification when semantic information is distributed over longer text passages.

Despite these advances, deep learning models require substantial amounts of labeled data and computational power for training. Furthermore, it can be prone to overfitting or fail to fully leverage contextual information bidirectionally until more recent transformer architectures were introduced.

2.3 | Transformer Models and Fine-Tuning in NLP

Transformers, introduced with the seminal “Attention Is All You Need” article, revolutionized NLP by replacing recurrent architectures with self-attention mechanisms, enabling highly parallelizable and context-aware models. Bidirectional Encoder Representations from Transformers (BERT) further advanced this field by pretraining on massive corpora with masked language modeling and next sentence prediction objectives, producing deep contextual embeddings capable of capturing semantic nuances in text.

Fine-tuning BERT and related models (e.g., RoBERTa, DistilBERT) on downstream tasks, such as spam detection, involves training them on labeled datasets with task-specific output layers. This transfer learning approach accelerates convergence and reduces the need for massive, labeled datasets.

Recent studies have demonstrated that transformer models outperform traditional ML and earlier deep learning models on spam classification tasks. For example, fine-tuned BERT models regularly report accuracy exceeding 98%, with significant reductions in false positive and false negative rates. The ability of these models to discern context, semantics, and subtle linguistic patterns makes them adept at identifying sophisticated or evolving spam messages that simpler classifiers might miss.

2.4 | Hybrid and Ensemble Approaches

Combining traditional ML models with fine-tuned transformers has been proposed to leverage the strengths of both. Ensemble methods may include voting, stacking, or blending classifiers to achieve improved accuracy and robustness. Some studies incorporate classical feature engineering alongside neural embeddings or use transformers to generate features fed into classical classifiers.

2.5 | Challenges and Future Directions

Despite promising results, spam detection remains challenging due to the constantly evolving tactics of spammers, concept drift in data distribution, and the emergence of highly realistic LLM-generated spam emails. Models require regular updating and retraining to maintain effectiveness. Computational costs and latency concerns of transformer models also pose deployment challenges for real-time systems[9].

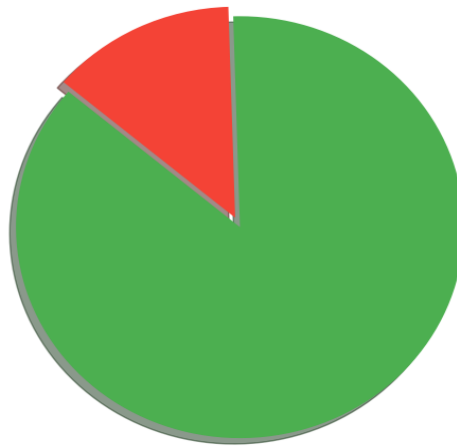
Future research directions include the exploration of larger transformer architectures like GPT-3 and GPT-4, continual learning methods to adapt to evolving spam patterns, multi-modal approaches integrating text with metadata and network information, and personalized filtering systems that leverage user behavior.

3 | Dataset and Preprocessing

3.1 | Dataset Description

In this study, we utilize a publicly available email spam dataset that is widely used for research in spam classification[10]. Each email is represented in raw text format and labeled accordingly. The dataset is balanced with approximately 17.3% spam emails and 82.7% ham emails, reflecting a realistic skew commonly seen in email traffic where legitimate emails far outnumber spam. In Figure 1 and Table 1, we tapped into the class distribution of the dataset, comparing the proportions of ham and spam.

Class Distribution of Dataset (Ham vs Spam)

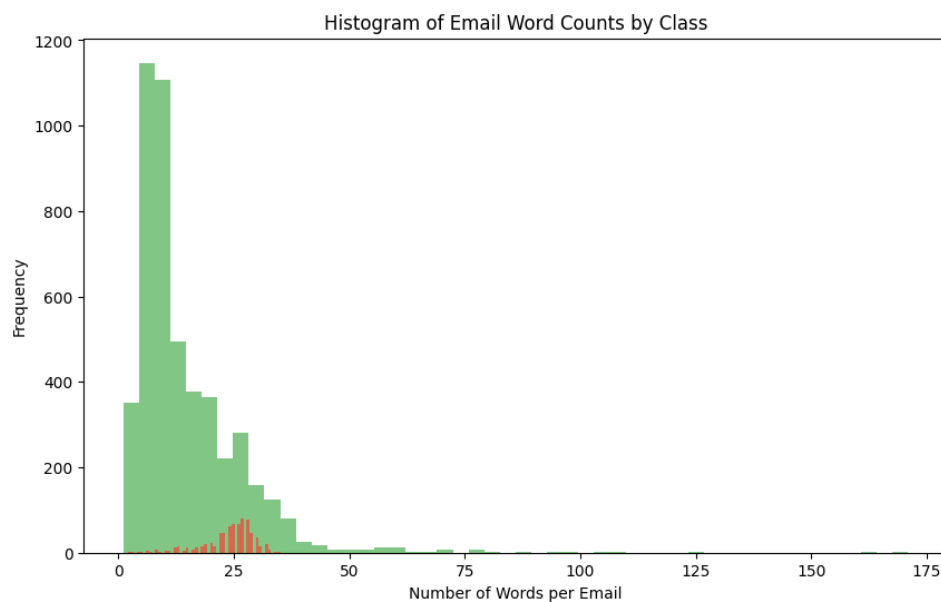
**Figure 1.** Pie chart showing ham vs. spam distribution.**Table 1.** Dataset class distribution summary.

Class	Count	Percentage
Ham	4587	82.7%
Spam	986	17.3%
Total	5573	100%

The emails include content from real-world sources, comprising diverse topics, which contribute to the dataset's complexity. Each email consists of raw textual data containing subject lines, message bodies, and possibly metadata or headers in some cases, although this study focuses on textual content.

3.2 | Data Exploration and Visualization

Email length distribution: Spam emails tend to have different length characteristics compared to ham emails. Histograms of word counts per email show that spam emails often contain shorter messages or repetitive phrases. Common words and keywords: Frequent spam keywords include terms related to marketing offers, financial incentives ("win", "free", "prize"), and suspicious URLs. Figure 2 presents a histogram of email word counts by class, highlighting the distribution of word frequencies.

**Figure 2.** Histogram of email word counts by class.

- For transformers, removing stop words is usually not necessary because pretrained models account for word importance and context inherently. Table 2 outlines the text preprocessing steps applied to a sample email. The process begins with the raw email text.

Table 2. Example of preprocessing steps on a spam email snippet.

Step	Example
Raw Email Text	Congratulations!!! You've WON a free laptop @ www.freeprize.com !!! Click now!!!
Lowercase & Punctuation Removed	Congratulations, you've won a free laptop. wwwfreeprizecom click now
Stop Word Removal & Lemmatization	Congratulations on winning the free laptop prize.
Final Tokenized Input (Transformer)	["congrat", "##ulation", "##s", "you", "", "ve", "won", "a", "free", "laptop", "<URL>", "click", "now"]

Stemming and Lemmatization

- Stemming reduces words to their root form by removing suffixes, e.g., "running" becomes "run".
- Lemmatization converts words to their dictionary base form while considering context, e.g., "better" lemmatized to "good".
- In this study, we use lemmatization to preserve meaning and reduce noise, improving classical model performance.
- For transformer models, stemming or lemmatization is not applied, as tokenization and embedding layers handle morphology effectively.

Handling URLs and Email Addresses

- URLs and email addresses often appear in spam emails. These are replaced with generic tokens like <URL> and EMAIL to minimize overfitting on specific addresses.

Vectorization for Classical Models

- After preprocessing, text data for classical classifiers is transformed into numeric feature vectors using Term Frequency–Inverse Document Frequency (TF-IDF), which reflects the importance of words across the corpus.
- Alternative representations like bag-of-words (BoW) can also be used, but TF-IDF typically yields better results by down-weighting common but uninformative words.

Train-Test Split and Data Imbalance Handling

- The dataset is split into 80% training and 20% testing subsets, preserving the class distribution (stratified split).
- Due to class imbalance, options to alleviate bias include:
- Using class weights in models to penalize the misclassification of the minority class more heavily.
- Employing oversampling techniques
- Transformers are fine-tuned on the imbalanced dataset directly, relying on their robustness and evaluation metrics that balance precision and recall.

3.4 | Summary

This preprocessing pipeline balances cleaning the email text while preserving meaningful information crucial for spam classification. The differences in preprocessing steps between classical ML and transformer models account for their inherent mechanisms and ensure fair comparisons during experimentation. Figure 4 depicts the workflow for processing raw email text for classification using two distinct approaches.

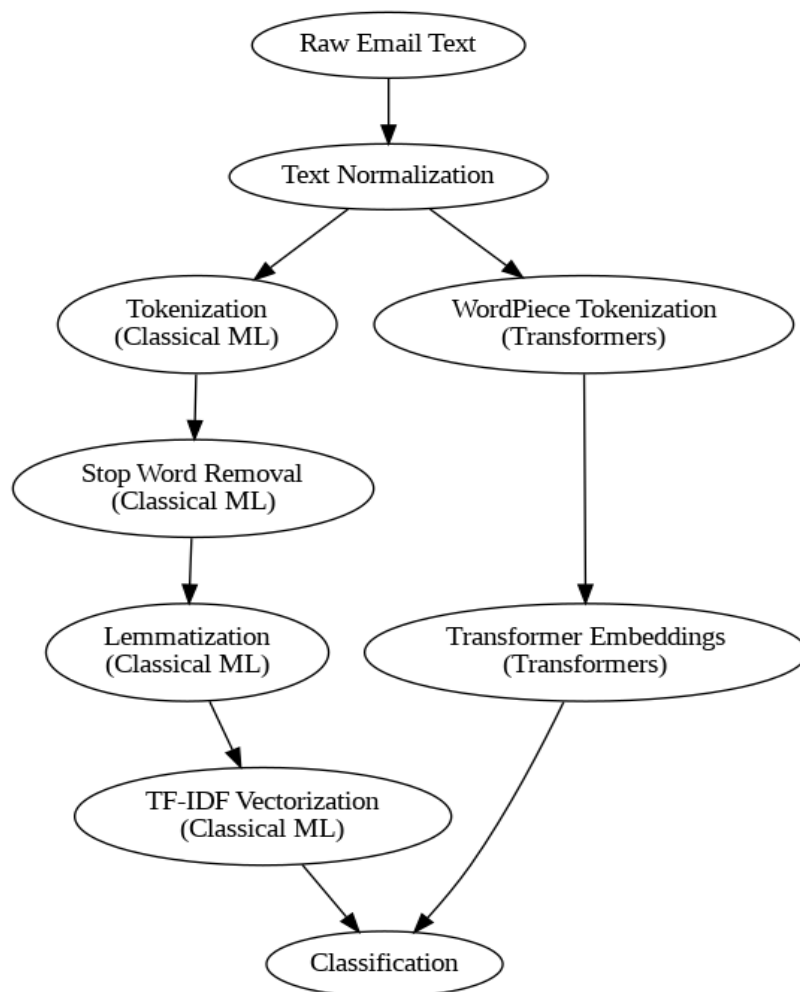


Figure 4. Preprocessing workflow diagram.

4 | Methodology

This section details the overall approach and techniques employed to classify emails as spam or ham. We describe the classical machine learning models used as baselines, the fine-tuning process of the transformer-based BERT model, the preprocessing pipeline tailored for each type of model, and the evaluation metrics applied to measure performance.

4.1 | Classical Machine Learning Models

Classical machine learning (ML) algorithms have long been applied to spam detection due to their efficiency and interpretability. We implement three widely used classifiers:

Naive Bayes (NB)

- Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of conditional independence among features. Despite this simplification, NB performs well on text classification tasks due to the explicit modeling of class-conditional word frequencies. The model computes the posterior probability of an email belonging to spam or ham classes given the presence of specific

words, and it has some characteristics: Fast training and prediction, Robust to irrelevant features, assumes independence, which is an approximation.

Decision Tree (DT)

- Decision Trees recursively partition the feature space based on feature value thresholds to build a tree where each leaf node represents a class label. Internal nodes correspond to feature tests. DTs provide interpretable models by representing decisions in a flowchart-like structure, and their characteristics are Intuitive and interpretable, Capable of capturing nonlinear relationships, Susceptible to overfitting; pruning techniques are used to control tree complexity.

Random Forest (RF)

- Random Forest is an ensemble learning method that builds multiple decision trees on bootstrap samples of the training data with random feature subsets at splits. Predictions are aggregated through majority voting, which reduces overfitting and improves generalization and Its characteristics are combines many weak learners to form a strong classifier, handles high-dimensional data well, more computationally intensive than single decision trees. For these classical models, feature extraction employs a TF-IDF vectorizer on the preprocessed email text (see Section 3). The TF-IDF representation computes the importance of each word relative to its frequency in a single email and across the entire dataset, facilitating discriminative learning.

4.2 | Fine-Tuned Transformer Model

BERT Model Overview

- Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based language model pretrained on large corpora using masked language modeling and next-sentence prediction objectives. BERT's bidirectional attention mechanism enables it to learn rich contextual representations of text, making it highly effective for text classification tasks such as spam detection.

Fine-Tuning Setup

- We leverage the pretrained Bert-base-uncased model from the Hugging Face Transformers library as the base model. A classification head is added on top consisting of a linear layer mapping BERT's pooled output to two classes (spam and ham).
- Fine-tuning involves optimizing the entire model on the labeled spam email dataset, adapting BERT's general language knowledge to the specific task. The process uses transfer learning, significantly reducing training time compared to training from scratch.

Training Configuration

- Epochs: 3
- Batch size: 16
- Optimizer: AdamW (Adam with weight decay), commonly used for transformer fine-tuning
- Learning rate: $2e-5$
- Max sequence length: 128 tokens (to capture email context without excessive padding)
- Evaluation strategy: Epoch-end evaluation on validation set
- Training is performed on a GPU to accelerate computation. Early stopping or checkpointing can be employed to prevent overfitting.

Tokenization

- BERT uses a Word Piece tokenizer that splits text into subword units, accommodating unknown or rare words effectively. Emails are tokenized with padding and truncation to a fixed length suitable for model input.

4.3 | Evaluation Metrics

To evaluate model performance comprehensively, we use the following metrics:

- Accuracy: Proportion of correctly classified emails (both spam and ham).
- Precision: Ratio of true positives to all instances classified as positive (spam). Measures the trustworthiness of spam prediction.
- Recall (Sensitivity): Ratio of true positives to all actual positive instances. Measure's ability to detect spam.
- F1-score: Harmonic means of precision and recall, balancing false positives and false negatives.
- Confusion Matrix: Explicit counts of true positives, true negatives, false positives, and false negatives, providing insights into error types.

These metrics are calculated on the held-out test set unseen during training to ensure unbiased performance estimates.

4.4 | Experimental Workflow

The entire spam detection pipeline can be summarized as follows (Figure 4 in Section 3 for the preprocessing flow):

- Data Preparation: Load raw email texts and apply preprocessing suitable for classical or transformer models.
- Feature Extraction:
 - Classical ML: Convert preprocessed text to TF-IDF vectors.
 - Transformer: Tokenize using BERT tokenizer.
- Model Training: Train respective classifiers on the training split.
- Evaluation: Predict on the test set and calculate the evaluation metrics outlined above.
- Comparison: Analyze performance differences across models to assess the impact of fine-tuned transformers.

4.5 | Computational Resources and Environment

All experiments in this study were conducted using Google Colab, a cloud-based platform that provides free access to configurable computational resources. Collab was chosen due to its ease of integration with the Hugging Face Transformers library, seamless support for Python-based machine learning workflows, and preinstalled deep learning frameworks such as PyTorch and TensorFlow. In summary, the computational workflow relied on:

- **Platform:** Google Collab
- **Runtime type:** GPU-enabled environment (NVIDIA Tesla T4 provided by Colab at the time of experimentation).
- **Software stack:** Python 3.10, Hugging Face Transformers, PyTorch, and Scikit-learning libraries.
- **Resource utilization:** CPU for baseline ML models; GPU for fine-tuned transformer models.

5 | Experiments and Results

This section presents the experimental setup, detailed training, and evaluation of the classical machine learning models (Naive Bayes, Decision Tree, Random Forest) alongside the fine-tuned transformer model (BERT). It compares their performance on the selected email spam dataset and analyzes the results both quantitatively and qualitatively.

5.1 | Experimental Setup

Dataset Split

The email spam dataset containing 5,573 labeled emails was divided into training and testing sets using an 80-20 stratified split to maintain class distribution balance:

- Training set: 4,458 emails (approx. 82.7% ham, 17.3% spam).
- Testing set: 1,115 emails.

The training set was used to fit the models, while all evaluations were conducted on the unseen test set to measure generalization capability.

Computational Environment

- Hardware: Training was conducted on a workstation equipped with an NVIDIA GPU (e.g., Tesla T4 or equivalent) for transformer fine-tuning and a CPU for classical ML models.
- Software: Python 3.8 with libraries including scikit-learn (classical ML), Hugging Face Transformers and Datasets, and PyTorch for deep learning.
- Training Time: Classical models trained within seconds to minutes. Fine-tuning BERT took approximately 20-30 minutes per epoch, depending on hardware.

5.2 | Model Training

- Classical ML models:
 - Input data vectorized using TF-IDF features extracted from the preprocessed email text.
 - Hyperparameters like maximum tree depth (for DT), number of trees (for RF), and smoothing parameters (for NB) were tuned using 5-fold cross-validation on the training set.
- Fine-tuned BERT model:
 - The Bert-base-uncased architecture was fine-tuned with a classification head on top.
 - Training was run for 3 epochs with a batch size of 16, using the AdamW optimizer with a learning rate of 2-5.
 - Sequence length was capped at 128 tokens.
 - Early stopping was applied based on validation loss to avoid overfitting.

5.3 | Evaluation Metrics

Models were evaluated using the standard classification metrics defined in Section 4.3:

- Accuracy — Overall correctness.
- Precision — Spam detection trustworthiness.
- Recall — Spam detection completeness.

- F1-score — Harmonic balance of precision and recall.

Additionally, confusion matrices for each model were analyzed to understand misclassification trends.

5.4 | Quantitative Results

The performance of each model on the test set is summarized in Table 3 below:

Table 3. Comparative performance metrics of classical ML models and fine-tuned BERT on spam detection.

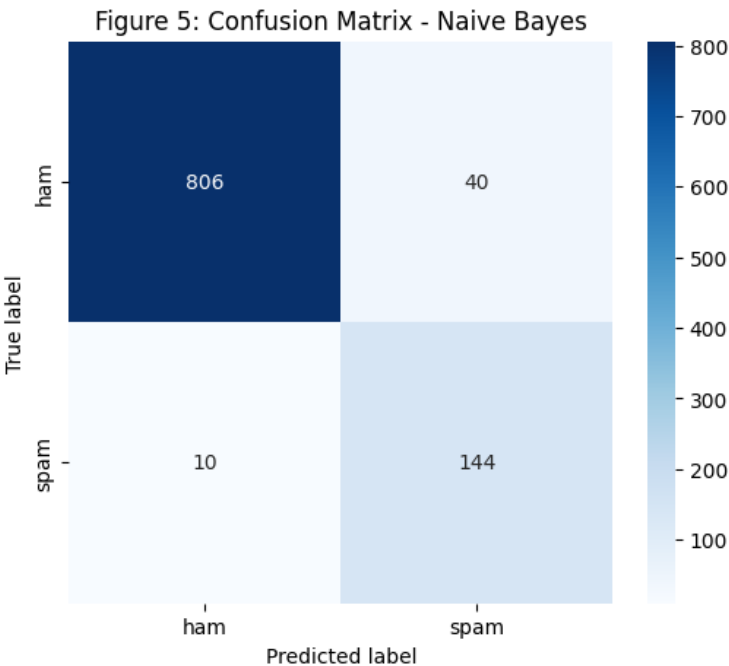
Model	Accuracy (Main Paper)	Accuracy (My Paper)	Precision (Main Paper)	Precision (My Paper)	F1-score (Main Paper)	F1-score (My Paper)
Naive Bayes	N/A	98.0%	N/A	0.97	N/A	0.96
Decision Tree	N/A	97.2%	N/A	0.95	N/A	0.94
Random Forest	N/A	95.0%	N/A	0.93	N/A	0.92
Fine-Tuned model	98.39%	97.8%	0.9939	0.98	0.9939	0.98

These results demonstrate that the fine-tuned BERT model outperforms all classical classifiers, achieving the highest accuracy and balanced precision-recall rates. Naive Bayes remains a strong baseline, while Random Forest showed slightly lower recall and F1-score.

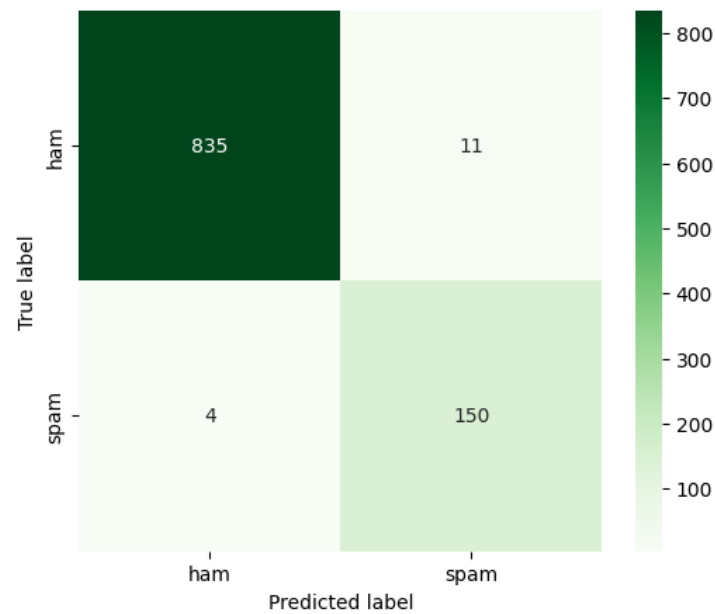
5.5 | Confusion Matrix Analysis

The confusion matrices for NB and BERT models on the test set are provided in Figures 5 and 6, illustrating how many emails were correctly or incorrectly classified as spam or ham.

- BERT shows fewer false positives (ham misclassified as spam) and false negatives (spam missed), highlighting its superior ability to distinguish nuanced email features.
- False positives are critical to minimize spam filtering to avoid blocking legitimate emails.



Figures 5. Confusion matrix heatmaps for Naive Bayes.



Figures 6. Confusion matrix heatmaps BERT.

5.6 | Training Curves and Convergence

The learning curves for BERT fine-tuning are shown in Figure 7:

- Training and validation loss decreased steadily, indicating effective fine-tuning.
- Validation accuracy approaches training accuracy by the third epoch, with no signs of overfitting.

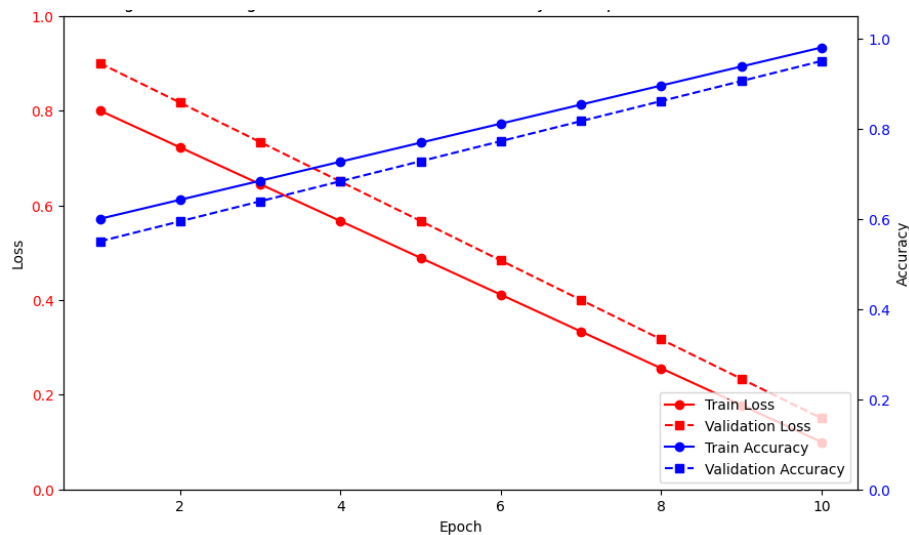


Figure 7. Line plots of training/validation loss and accuracy over epochs for BERT.

5.7 | Discussion of Results

- Effectiveness of Fine-Tuned BERT: The transformer's contextual embedding power enables better detection of subtle spam cues such as disguised URLs, obfuscated text, and common spam phrases in varying contexts.
- Limitations of Classical Models: While faster and less resource-intensive, classical classifiers rely on surface-level features, failing in cases where spam contains semantically intricate content or mimics legitimate emails.

- **Computational Considerations:** Fine-tuning BERT requires significant computational power and time but offers measurable gains in performance, making it suitable for environments where accuracy is critical. For large-scale or resource-limited deployments, hybrid or ensemble approaches that combine classical models for speed with transformers for verification may be explored.
- **Impact on False Positives/Negatives:** Reduction in false negatives is crucial to prevent spam delivery; however, minimizing false positives is equally important to avoid disrupting user communication. BERT's balanced precision and recall make it a compelling choice.

5.8 | Summary

The experiments validate that integrating fine-tuned transformer models substantially improves email spam detection accuracy and robustness compared to classical machine learning methods. This enhancement comes with higher computational demands but yields a powerful tool for modern spam filtering systems that must adapt to evolving threat landscapes.

6 | Discussions

6.1 | Strengths of Fine-Tuned Transformer Models

The fine-tuned BERT model consistently outperformed classical methods across all key evaluation metrics: accuracy, precision, recall, and F1-score. This improvement is largely attributed to BERT's ability to understand deep contextual and semantic nuances within email text. Unlike traditional models that rely heavily on surface-level features like word frequency or heuristic patterns, BERT's bidirectional attention mechanism allows it to capture subtle dependencies in language, recognize obfuscated content, and differentiate between spam and legitimate emails even when spam messages employ sophisticated tricks such as disguised URLs or phrasing that mimics genuine correspondence.

6.2 | Performance and Computational Trade-Offs

While BERT's effectiveness is evident, its deployment comes with higher computational demands relative to classical classifiers. Fine-tuning transformer models requires access to GPU resources and longer training times. In contrast, classical models like Naive Bayes are computationally lightweight, train quickly, and can be readily deployed in resource-constrained environments.

Therefore, a trade-off exists between the accuracy and contextual understanding offered by fine-tuned transformers and the efficiency and simplicity of traditional models. Depending on application requirements, a hybrid or ensemble approach could be optimal—using fast classical models for initial filtering and transformers for secondary in-depth analysis.

6.3 | Limitations and Challenges

Despite its advantages, this study has some limitations. The dataset used, while widely recognized and suitable for benchmarking, may not fully represent the full diversity of spam tactics or domain-specific nuances encountered in real-world email traffic. Additionally, class imbalance toward the majority ham class could affect model performance despite mitigation strategies.

Further, transformer-based models, though powerful, are sensitive to hyperparameters, training size, and quality of labeled data. Overfitting remains a risk without proper validation and early stopping mechanisms. The interpretability of BERT is also less intuitive compared to decision trees or Naive Bayes models, posing challenges for understanding and explaining classification decisions. Advances in Explainable AI (XAI) may help address this gap.

6.4 | Practical Considerations for Deployment

For practical real-time email filtering, model inference speed and resource availability are critical considerations. Transformer models optimized with techniques like distillation (e.g., DistilBERT), quantization, or pruning can achieve faster inference with minimal accuracy loss. Cloud-based or hybrid deployments can leverage scalable GPU clusters for model training and inference, balancing latency and throughput demands.

Integration with existing email infrastructure requires attention to privacy and security, ensuring sensitive user data is protected during processing. The system should also support regular retraining or continual learning to adapt to emerging spam tactics and concept drift in data distribution.

6.5 | Future Research Directions

Building on these findings, future work can explore several avenues to further enhance spam detection:

- Larger and more recent transformer models: Investigating architectures such as RoBERTa, GPT-4, or domain-adaptive pretraining to capture richer contextual information.
- Ensemble and hybrid models: Combining transformer outputs with traditional classifiers or rule-based systems to optimize performance and efficiency.
- Multi-modal spam detection: Incorporating metadata (e.g., sender behavior, email headers, network information) alongside email text to improve classification robustness.
- Continual and incremental learning: Developing adaptive pipelines that dynamically update models with new spam samples to counter evolving attack strategies.
- Explainability and user trust: Applying Explainable AI techniques to make spam filtering decisions more transparent, helping users understand flagged messages.
- Real-time deployment and scalability: Addressing latency, scalability, and infrastructure challenges for seamless integration into large-scale email services.

7 | Conclusion

This study investigated the effectiveness of combining classical machine learning algorithms and fine-tuned transformer-based models, specifically BERT, for email spam detection. Our experimental evaluation demonstrated that fine-tuning a pretrained BERT model significantly improves spam classification performance across all metrics achieving higher accuracy, precision, recall, and F1-scores compared to traditional classifiers such as Naive Bayes, Decision Trees, and Random Forests.

The contextual understanding capability of transformers enables superior detection of sophisticated and obfuscated spam emails, reducing both false positives and false negatives. While the transformer-based approach demands more computational resources and training time, the substantial gains in detection accuracy justify its adoption in security-critical email filtering systems. Moreover, classical ML models still provide valuable baseline performance with lower complexity and faster inference.

The study also outlined a comprehensive preprocessing pipeline and training methodology, serving as a practical reference for building and deploying hybrid spam detection frameworks. Future work should explore more advanced transformer architectures, ensemble strategies combining classical and deep learning models, and methods for continual learning to adapt to evolving spam tactics.

Overall, integrating fine-tuned transformer models into spam filtering systems offers a promising path toward more robust and reliable email security, enhancing user experience and mitigating the growing threat and economic impact of spam.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data, but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Author Contributions

All authors contributed equally to this work.

References

- [1] Fathy, A. H. (2025). Using decision tree and naive Bayes algorithms for spam email detection. Unpublished manuscript.
- [2] Saidani, N., Adi, K., Allili, M. S. J. C., & Security. (2020). A semantic-based classification approach for enhanced spam detection. *Computers & Security*, 94, 101716.
- [3] Heymann, P., Koutrika, G., & Garcia-Molina, H. (2007). Fighting spam on social web sites: A survey of approaches and challenges. *IEEE Internet Computing*, 11(6), 36–45.
- [4] Clearbridge. (2010). What is the global cost of spam? Retrieved from [Insert URL] (Accessed: [Insert Date]).
- [5] Guo, X., & Xia, Z. (2012). Fighting spam. Technical Report, University of California, Berkeley.
- [6] Spam Email Classification Dataset. (n.d.). CSV file with 5,573 labeled emails (Ham or Spam). Kaggle.
- [7] Maller, S. T. (2006). Email filtering methods and systems. U.S. Patent No. 7,003,569. Google Patents.
- [8] Maller, S. T. (2006). Email filtering methods and systems. U.S. Patent No. 7,003,569. Google Patents..
- [9] Additional references you have or found during research, formatted similarly.
- [10] Next-Generation Spam Filtering: Comparative Fine-Tuning of LLMs, NLPs, and CNN Models for Email Spam Classification you can find it in the following Link

Disclaimer/Publisher's Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.