iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74310

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Email Spam Detection: Comparative Review of Deep, Transformer, and LLM Techniques

Ajay Chani[1], Jai Bhagwan[2]
[1]*M.Tech. Student, Department of CSE, Guru Jambheshwar University of Science & Technology, Hisar - Haryana (India)*
[2]*Assistant Professor, Department of CSE, Guru Jambheshwar University of Science & Technology, Hisar - Haryana (India)*

Abstract: ***Email remains a mission-critical medium for personal and enterprise communication—and a persistent vector for abuse. Modern spam ranges from bulk advertisements to sophisticated social engineering and phishing, creating burdens on users and infrastructure while elevating risk. Recent advancements have shifted research decisively from sparse, hand-engineered features to deep neural architectures and transformer-based encoders; the latest wave introduces large language models (LLMs) with zero/few-shot classification and stronger semantic understanding. This review synthesizes peer-reviewed developments across that spectrum, comparing representative approaches and highlighting consistent themes: transformers generally outperform classical baselines on content-centric tasks; deep and graph-augmented models capture longer-range and relational cues; and LLMs enable intent-aware classification but pose cost, latency, and governance challenges. We identify open problems around dataset shift, multilingual and code-mixed content, explainability, measurement realism, and production constraints. We conclude by recommending hierarchical spam classification—a two-stage pipeline that first filters spam and then semantically categorizes legitimate mail—as a practical direction to improve downstream triage and analyst productivity while retaining precision at the perimeter.***

Keywords: Email spam detection, Hierarchical email classification, Transformers (BERT/DistilBERT), Large language models (LLMs), Zero-shot/few-shot classification.

I. INTRODUCTION

Email remains a cornerstone of digital communication for billions of users, serving critical personal and business functions. Unfortunately, unsolicited bulk email (spam) has risen in step with email's ubiquity—accounting for an estimated 50–85% of global email traffic in recent years [1]. Far from a mere nuisance, spam now frequently carries phishing scams, malware, fraud schemes, and other security threats [1], [2]. The intersection of scale and intent underscores the necessity of effective spam detection in safeguarding users' privacy and resources. Indeed, various industry reports indicate that hundreds of billions of spam messages are dispatched each day, presenting a significant challenge to cybersecurity efforts [1]. In response to the overwhelming influx of spam emails, researchers and practitioners have actively sought out more sophisticated filtering techniques to address this challenge [1].

Early Approaches – Rule-Based Filtering: In the initial stages of tackling spam, the strategies employed were largely based on a foundation of knowledge and comprehension. Email providers and administrators relied on thoughtfully crafted rules, keyword blacklists and whitelists, along with heuristic if-then filters to effectively identify and manage spam [2]. SpamAssassin and similar systems assess messages by allocating points to identified spam phrases or sender domains, ultimately preventing those that exceed a specific threshold from reaching the inbox. While these rule-based filters were simple and sometimes effective, they exhibited a certain fragility—spammers could easily circumvent the established rules by camouflaging text (for example, writing "viagra" as "v1@gra") or modifying sender addresses. Moreover, maintaining a thorough set of regulations became a challenging endeavour, frequently resulting in errors. The shortcomings of static rule and blacklist methods became clear as they found it increasingly difficult to adapt to the constantly changing strategies employed by spammers, prompting a shift towards more dynamic, data-driven approaches [2]. Researchers recognised the importance of developing a strategy that enhances learning, with the goal of automatically adapting to new spam patterns while reducing the likelihood of misclassifying legitimate (ham) emails as spam.

Classical Machine Learning Era: Machine learning (ML) methods were the most often used paradigm for spam detection by the early 2000s. Unlike hard-coded rules, filters based on machine learning develop the ability to classify emails by being trained on extensive collections of labelled examples. Pioneering studies demonstrated the superior accuracy of statistical text classifiers—most famously the Naïve Bayes probabilistic model—for filtering spam [2]. Soon a wide range of supervised learning algorithms were applied, including support vector machines (SVM), decision trees, K-nearest neighbors, and ensemble methods [2], with ongoing work on feature selection and dynamic updates to sustain performance under drift.

©IJRASET: All Rights are Reserved

These approaches encoded emails as feature vectors (e.g., word frequencies, n-grams, sender metadata) and achieved ever-improving performance. For instance, Dada, Bassi, Chiroma, Abdulhamid, Adetunmbi, and Ajibuwa (2019) reviewed numerous ML-based filters reporting over 90% spam detection accuracy across different datasets [2], [1]. In real-world deployments, major email providers (Gmail, Yahoo, Outlook) adopted continuously trained ML models to stop spam before it reaches user inboxes [1]. The learning-based paradigm thus significantly advanced spam filtering effectiveness.

However, even as classifiers grew more accurate, spammers adapted in parallel—employing tricks like gibberish text, image-based spam, and domain hopping to foil detection [1]. This cat-and-mouse dynamic meant that no single static model could completely eliminate spam. The arms race between spam authors and filter developers necessitated constant innovation in algorithms and features [1].

Deep Learning and Hierarchical Models: Researchers have been using deep learning more and more in the past ten years. Deep learning is a type of multi-layer neural network that can automatically find complicated characteristics. This has helped spam detection work even better. Initial research utilised basic feed-forward neural networks for email filtering, occasionally using balancing techniques to mitigate the disparity between spam and legitimate categories [3]. This area of work rapidly grew to include architectures that were focused on text. Convolutional Neural Networks (CNNs) were used to find short-range lexical patterns, while Recurrent Neural Networks (RNNs) were used to capture the flow of language across time. CNNs were beneficial for more than only finding text-based spam; they could also find spam hidden in images by considering text in images as visual characteristics [3].

LSTMs and GRUs, which are examples of recurrent variations, were better at simulating longer passages and more complex word dependencies. Zavrak and Yılmaz (2023) used a CNN-based feature extraction method with a GRU sequence model and an attention layer to make the system focus on the most important sections of an email [4]. These hybrid neural techniques have repeatedly demonstrated great accuracy, frequently exceeding 95%, while enhancing generalisation through layered representations of email content [4].

Researchers also started looking on hierarchical classification pipelines about the same time. Doshi, Parmar, Sanghavi, and Shekokar (2023) presented a two-tier paradigm that initially differentiated spam from ham and subsequently subdivided the spam category into phishing and non-phishing variants [5]. Researchers have also looked at clustering-based methods that group spam by subject or intent before classifying it. This lets them use different models for each group [1]. Hierarchical or multi-stage systems provide enhanced granularity by distinguishing between really malicious phishing efforts and relatively innocuous advertising emails, hence facilitating more accurate and adaptive screening.

Transformer and LLM Advances: Recent advancements in NLP, mostly attributed to Transformer encoders, have significantly enhanced spam filtering. Fine-tuned BERT models have become good starting points for categorising email because its bidirectional attention shows how the meaning of a word changes with its neighbours, something bag-of-words features don't do [6]. In practice, this contextual approach enables a filter perceive cues like a "verify your account" request as suspicious only when the wording around it suggests that it is trying to steal your information. According to empirical investigations, adopting BERT representations improves both accuracy and recall. Tida and Hsu (2022) presented a deployed version that worked well in real time while also capturing spam [6]. Building on this concept, Zouak, El Beqqali, and Riffi (2025) integrated BERT text embeddings with a GraphSAGE layer, enabling the model to analyse associations among emails, such as shared senders or temporal closeness. That hybrid captured signals beyond just content and did very well on a number of benchmarks, showing how important it is to model content and structure together [7].

At the same time, big language models (LLMs) have opened up a new way. You may provide systems like GPT-3/ChatGPT a few instances and ask them to sort mail without any training for that specific purpose [8]. Initial assessments are inconclusive: Wu, Si, Zhang, Gu, and Wosik (2024) noted that in-context ChatGPT lags behind tuned deep classifiers on a large English corpus but outperforms BERT on a smaller Chinese dataset—indicating potential utility for zero-/few-shot applications in low-resource environments, despite LLMs not yet serving as a comprehensive substitute for specialised filters [8]. Looking future, real-world pipelines will probably use a mix of these tools: little Transformers for quick, first-pass screening and selected LLM reasoning when a message is unclear.

Overall, the field has moved from hand-crafted rules to classical ML, then to deep networks and Transformers, and is now testing LLMs for the hardest cases. Each phase has increased the system’s ability to generalize and interpret context, and the remainder of this review compares these strands and shows how they can be combined in hierarchical pipelines to strengthen email security.

II. LITERATURE REVIEW

A. *Foundational and Classical Techniques*

In the early days of spam filtering, the methods primarily relied on rule-based and list-driven techniques. These approaches involved assigning scores to carefully selected indicators, such as suspicious keywords, anomalies in headers, and hits on blacklists, ultimately leading to the blocking of messages that surpassed a certain threshold. Systems inspired by SpamAssassin illustrate this concept well: they are designed to be transparent and straightforward to adjust, yet they ultimately prove fragile as spammers evolve through techniques such as obfuscation (including misspellings and encoding tricks), rapidly changing domains, and header spoofing [2], [9]. Maintaining these rulesets was a significant challenge, as their inflexibility resulted in both the failure to catch spam (false negatives) and the unintended blocking of legitimate emails (false positives). This issue became more pronounced as campaigns expanded to include various languages and templates [2], [9]. The pressures led to a transition towards learning methods that are informed by data and can extend beyond established patterns [10].

In the realm of classical machine learning, two families have emerged as particularly significant: Naïve Bayes (NB) and Support Vector Machines (SVM). Naive Bayes utilised straightforward probabilistic principles to convert token frequencies into efficient, lightweight classifiers that performed well with large volumes of email traffic—frequently the go-to option when resources or time constraints were a concern [2], [10]. Support Vector Machines (SVMs) demonstrate a capacity to establish more defined decision boundaries within high-dimensional spaces, such as those involving character or word n-grams. This capability results in enhanced precision and resilience when dealing with overlapping classes, although it does come with the trade-off of increased demands during both training and inference processes. [10], [11]. In practical applications, the process of feature engineering—including decisions around tokenisation, the selection of n-gram ranges, and the implementation of weighting schemes such as TF-IDF, along with the strategic use of metadata like sender/domain and basic reputation signals—proved to be as significant as the learning algorithm itself. Additionally, it was common for pipelines to incorporate normalisation and dimensionality reduction techniques to manage sparsity [10]. While these models demonstrate strong performance in benchmarks and are easy to deploy, they exhibit certain structural limitations. These include a dependence on superficial lexical cues, a vulnerability to shifts in underlying concepts, a restricted ability to grasp long-range semantics, and a propensity to analyse each message in isolation. The presence of these gaps, along with the increasing sophistication of adversarial techniques, has paved the way for representation-learning methods. This includes deep neural networks and, subsequently, transformers, which are capable of learning more nuanced context directly from text, thereby alleviating the need for extensive hand-crafted features [10].

B. *Deep Learning Techniques*

The rise of deep learning introduced models that learn directly from raw email text rather than relying on hand-crafted indicators. Sequence-aware architectures became especially influential: Convolutional Neural Networks (CNNs) capture short, local patterns (e.g., phrase motifs and orthographic variants), while Recurrent Neural Networks (RNNs) and their gated forms (LSTM, GRU) track longer narrative flow and dependencies that often distinguish persuasive phishing from legitimate correspondence. With minimal preprocessing and suitable embeddings, these networks build layered representations of messages and operate effectively on subject and body text alike.

Across comparative studies, deep models generally outperform classical ML and lexicon/rule systems for spam filtering [10]. Their edge comes from learning non-linear interactions among tokens and context, allowing cues like “free” to be judged by nearby words (e.g., “free money” vs. “free speech”) rather than as a standalone trigger [10]. Empirical results echo this: Rafat et al. (2022) reported that a deep neural network reduced both false positives and false negatives relative to an SVM baseline [13]. Likewise, Saleem et al. showed that an LSTM-based filter recovered spam that Bayesian methods missed by leveraging richer sequence context [14]. In practice, these models are particularly strong on nuanced, socially engineered spam where surface keywords are insufficient.

A notable step forward is combining network types to balance accuracy and efficiency. For instance, Saleem et al. fused LSTM (for long-range context) with GRU (for lighter computation), achieving high detection scores on Enron (~90% accuracy; AUC ≈ 98.99%) while mitigating issues like vanishing gradients and excessive runtime [14]. The broader lesson is architectural pragmatism: very deep or over-parameterized models can overfit modest datasets or prove too slow for high-throughput mail streams, so researchers often employ regularization, batch normalization, and carefully chosen depths to keep systems deployable. Despite open challenges—data hunger across domains/languages and limited interpretability that can complicate audits—deep learning now underpins many modern email filters thanks to its consistent gains on hard-to-detect spam [10], [13], [14].

C. Transformer-Based Models (BERT and Variants)

The introduction of transformer models, which are based on the success of deep learning, has greatly improved the ability to find spam. Transformers, like BERT (Bidirectional Encoder Representations from Transformers), use self-attention processes to understand how words in a text are related to one other. Unlike RNNs, which only look at one token at a time, transformers look at all of them at once and use self-attention to model associations throughout a whole phrase (or document) at the same time. Their depictions are extremely rich because they see the whole picture. BERT, RoBERTa, and DistilBERT are large pretrained encoders that are initially trained on huge collections of generic text and then fine-tuned on smaller, task-specific datasets. This transfer arrangement has been quite successful for spam filtering.

Tida and Hsu (2022) improved BERT and found that it worked well across diverse types of spam, which supports the premise that one contextual model may work for all types of spam [6]. BERT embeddings capture sense in context, which means that words like "secure" mean different things in phrases like "secure account" and "secure transaction." This helps find phishing emails that are carefully phrased. Uddin et al. advanced this research by optimising DistilBERT (and evaluating associated transformer baselines), achieving state-of-the-art outcomes—such as about 99% accuracy on Enron/SMS tasks—and integrating the model with interpretative tools to highlight significant tokens [15]. In a related study, Nasreen et al. (2024) used BERT and a Grey Wolf Optimization-based selector to get rid of noisy inputs, which made decision boundaries clearer and cut down on computing time. They found that this worked better than simpler methods on LingSpam [12]. Together, these studies highlight the pattern: contextual transformers create a solid baseline for spam/phishing, and attentive fine-tuning or feature selection may push performance and efficiency further. In addition to standalone transformers, researchers have explored combining transformers with other techniques. A notable example is the work of Zouak et al. (2025), who integrated BERT embeddings with a GraphSAGE graph neural network to capture both textual and structural relationships in email data [7]. In their BERT-GraphSAGE model, each email is represented by a BERT-derived vector and treated as a node in a graph, where edges connect similar emails. GraphSAGE then learns to propagate and aggregate information between neighboring emails. This hybrid achieved 96–99% accuracy on multiple benchmark datasets (Enron, SpamAssassin, LingSpam), outperforming several state-of-the-art baselines [7]. The success of such models indicates that transformer-based semantic understanding, when paired with complementary methods (like graph networks capturing inter-email patterns or meta-data), can yield highly robust spam classifiers.

While transformer models clearly offer superior accuracy and a deeper comprehension of email content, they come with practical considerations. BERT and its ilk are resource-intensive; a base BERT model has hundreds of millions of parameters. Fine-tuning and inference can be slow without acceleration hardware, which is a concern for real-time spam filtering systems. For example, a distilled transformer model can process an email in a few milliseconds, but a larger or non-distilled model might take significantly longer, impacting scalability [16], [17]. There is often a trade-off between model size and speed. Techniques like model distillation (creating smaller models like DistilBERT) and quantization are thus important for deploying transformers in production spam filters [17]. Another limitation is the input length of transformers – standard BERT can only handle texts up to 512 tokens, which could truncate very long emails (though in practice, key spam cues usually appear early in the message). Despite these challenges, the trend in recent literature is to embrace transformers and manage their complexity, rather than shy away from them, because the payoff in accuracy and adaptiveness is substantial [10]. Many email providers now incorporate transformer-based models in their filtering pipeline, particularly for analyzing the email body content with high precision. We can expect continued refinement of these models (through better pretraining, fine-tuning strategies, and model compression) to make them faster and even more effective for spam detection tasks.

D. Large Language Models (LLMs like GPT)

The latest development in AI – large language models – promises to further revolutionize spam detection. LLMs such as OpenAI's GPT-3 and GPT-4 are trained on extraordinarily large text datasets and possess an impressive ability to understand and generate human-like text. These models, with billions of parameters, can perform a wide range of language tasks without task-specific training, using zero-shot or few-shot learning via prompting. In the context of spam detection, researchers are only beginning to explore LLMs, but initial findings are encouraging. The allure of LLMs is that they carry a vast amount of world knowledge and linguistic context, potentially allowing them to recognize spam by its subtle characteristics or anomalous context – for example, they might flag an email that "just doesn't read like" normal business correspondence even if specific spam keywords are absent. LLMs can also adapt on the fly: given a few example spam emails and ham emails in a prompt, a model like GPT-4 can immediately start classifying emails with no gradient-based training at all. This flexibility could enable rapid deployment of spam filters for new spam campaigns or languages where labeled data is scarce.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

One recent study evaluated a small LLM against a tuned transformer baseline on an email spam task and found that, while the LLM performed respectably, a fine-tuned DistilBERT achieved higher recall with far lower latency—highlighting the current efficiency gap between general LLMs and task-specific encoders [18]. Specifically, a distilled transformer achieved millisecond-level inference suitable for high-throughput filtering, whereas LLM inference was an order of magnitude slower [18]. This underscores a significant challenge: large language models require substantial computational resources, which complicates their application for real-time filtering of high volumes of email. Nevertheless, comparative assessments of ChatGPT-style prompting indicate that zero or few-shot classification can perform competitively on certain datasets, although it does not consistently outperform fine-tuned transformers [8].

Emerging concerns have arisen regarding the potential misuse of large language models (LLMs) by adversaries to create more convincing spam or phishing emails that could bypass traditional filtering systems [19]. This situation fosters a competitive dynamic in which defenders may increasingly depend on large language models to identify the distinctive indicators of AI-generated content or skilfully crafted spam. In conclusion, large language models signify a significant advancement in the field of spam detection research. Their exceptional language comprehension presents an opportunity to tackle some of the most challenging types of spam; however, the implementation of this technology must contend with considerable computational requirements. Current research is exploring methods to refine or tailor large language models for the purpose of spam classification, aiming to combine the advanced capabilities of GPT-scale models with the efficiency found in smaller models [18]. As LLM technology continues to evolve and improve, it is expected to significantly contribute to the development of next-generation anti-spam systems, particularly within the hybrid frameworks that will be discussed subsequently.

E. Hierarchical and Multi-Stage Classification Frameworks

No individual method has emerged as a definitive solution for the challenge of email spam detection. As a result, scholars and practitioners have crafted layered and multi-phase frameworks that integrate various approaches to enhance overall effectiveness. The concept involves harnessing the unique advantages of each method while addressing their limitations, resulting in a spam filtering pipeline that surpasses the individual contributions of its components. Today, many enterprise email filtering systems utilise a multi-layered approach to security. Emails navigate through a series of filters, which include IP and domain blacklists, signature-based detectors, and content-based classifiers. In their 2022 study, Ahmed and colleagues outline a typical multi-stage email filtering process utilised in various industries. Initially, basic content filters, occasionally enhanced by AI heuristics, assess the message. This is succeeded by header filters that scrutinise the metadata. Next, checks against blacklists and whitelists are performed, culminating in the application of rule-based filters that adhere to either user-defined or provider-specific guidelines [9]. An email must successfully navigate through all these stages without raising any flags in order to finally arrive in the inbox. This layered approach significantly minimises spam before the content progresses to the more demanding phases. They also provide defense-in-depth; even if one layer misses a spam, another might catch it.

In academic research, various hierarchical models have been proposed to systematically combine classifiers. One method involves a two-stage classification process. In the first stage, a straightforward and efficient classifier examines all emails. Following this, a more intricate model takes a closer look at those instances that remain uncertain or are likely to be spam. In their 2025 study, Stow and Ezonfa illustrate an effective approach with their two-stage spam filter. The first stage employs a lightweight Logistic Regression to swiftly eliminate clear spam, while the second stage delves deeper into the remaining emails using a feed-forward Neural Network, following a PCA-based feature reduction process [20]. Once an email progresses to Stage 2, the initial wave of straightforward spam—and even some uncomplicated legitimate messages—has been filtered out. This enables the neural network to concentrate on the more ambiguous cases, utilising a more nuanced set of features. This system demonstrated an impressive 98.0% accuracy in detecting spam within the SpamAssassin corpus and an even more remarkable 99.34% on LingSpam, showcasing a performance that surpasses that of each classifier individually [20]. The authors highlight that this hybrid successfully navigated the common tradeoff between speed and accuracy: the overall pipeline is both efficient, due to the rapid first stage, and remarkably precise [20]. Such results demonstrate the benefit of multi-stage design: the first stage cuts down the workload and noise, and the second stage provides a thorough analysis, resulting in a robust filter that is both fast and precise.

Another multi-stage strategy is hierarchical feature extraction using advanced neural architectures. Al-Kabbi et al. (2024) developed a hierarchical two-level model for SMS spam (which conceptually applies to email as well) that exemplifies this approach [21]. Their system first processes text at the word level using a CNN to capture local patterns, and at the same time processes at the sequence (sentence) level using a BiLSTM to capture context and word order [21]. These features are then fused by a Hierarchical Attention Network that learns to weight the most important words and sentences for the spam classification task [21].

The final classification layer uses this multi-level representation to decide spam or ham. By structuring the model hierarchically (words → sentences → message), it mirrors the linguistic structure of content and significantly improves detection of spam that may require understanding of both individual keywords and overall message context. The model achieved an impressive 99.48% accuracy on a benchmark SMS spam dataset [21], highlighting how hierarchical deep networks can boost performance by looking at the problem from multiple granularities. Although this example is in the SMS domain, similar ideas have been explored for email (for instance, treating an email as composed of subject and body, or splitting by paragraphs, and applying attention to fuse features). Such hierarchical deep models effectively embed a multi-stage process within one unified architecture.

Beyond these, other multi-stage and ensemble frameworks have been proposed: some works combine content-based classifiers with behavior-based filters (e.g., monitoring user email reading behavior or network-level sending patterns) in a pipeline, while others use voting or stacking ensembles of different algorithms (Bayes, SVM, neural network) to make a final decision. The common theme is to exploit diversity – different methods may catch different types of spam, so a combination can cover more ground. One challenge in multi-stage systems is ensuring that errors don't propagate (for example, if Stage 1 mistakenly drops a legitimate email, Stage 2 never gets a chance to save it). This is usually managed by setting conservative thresholds in early stages and letting later stages make the definitive call, or by allowing feedback loops where later stages can flag if they think an email was incorrectly handled upstream. Another consideration is complexity: multi-stage filters can be harder to maintain and require tuning of multiple components. Despite this, the literature shows a strong consensus that multi-stage and hierarchical approaches yield superior spam filtering performance. Recent reviews specifically recommend that all parts of an email (header, text, links, etc.) should be jointly considered to build a more robust spam detection framework [10] – essentially arguing for a holistic, multi-aspect analysis which is naturally achieved through a layered system.

Crucially, multi-stage frameworks provide a pathway to integrate cutting-edge models like LLMs into spam filtering in a practical manner. An example of a proposed hierarchical framework utilising LLMs could involve employing conventional machine learning or smaller deep learning models for the majority of emails, reserving the use of a large GPT-based analysis for messages that are either borderline or notably complex, such as targeted phishing attempts. In this manner, the overall system retains its efficiency while also gaining from the LLM's exceptional reasoning capabilities in complex situations. Certainly, the transition towards these hybrid frameworks is already in progress. The integration of rapid, specialised filters alongside more deliberate, general intelligent models represents a rational advancement in the ongoing battle against spam. In conclusion, hierarchical and multi-stage classification frameworks have emerged as fundamental components of contemporary spam detection systems. They provide improved accuracy, adaptability, and resilience by integrating the advantages of rule-based methods, classical machine learning, deep learning, and the latest transformer and large language model techniques into a unified defence strategy.

III. COMPARATIVE ANALYSIS OF SPAM EMAIL CLASSIFICATION MODELS

Table 1. Comparative Summary of Representative Spam Email Classification Studies

Study (Year)	Model(s) Used	Dataset	Reported Performance	Key Observations
Karim et al. (2019) [22]	Naïve Bayes, SVM, Random Forest (classical ML)	Multiple (survey of prior work)	~90–97% accuracy (various datasets)	Classical ML models perform well with feature engineering, but struggle with semantic nuances and adversarial spam.
Poobalan et al. (2024) [23]	Deep Neural Net (DNN-BiLSTM) vs. CNN	Enron email corpus	96.39% (BiLSTM); 98.69% (CNN accuracy)	Deep learning outperforms earlier methods – a CNN achieved 98.7% on Enron, surpassing traditional ML. Long email sequences still posed challenges for RNNs/CNNs.
Adnan et al. (2024) [24]	Ensemble (Stacking of LR, DT, KNN, NB, AdaBoost)	Combined spam corpora	98.8% accuracy (stacked ensemble)	An ensemble of heterogeneous classifiers outperformed individual models. Required data balancing and incurred higher computational overhead.

Nasreen et al. (2024) [12]	Hybrid GWO-BERT (BERT + Grey Wolf Optimizer) + CNN/LSTM/RF	Ling-Spam corpus	99.14% accuracy (GWO-BERT)	An optimized BERT-based model attained state-of-the-art accuracy on a single dataset. However, the complex hybrid approach was only evaluated on one corpus.
Tida & Hsu (2022) [6]	"Universal" BERT (pretrained, transfer learning)	Enron, SpamAssassin, Ling, etc.	97% accuracy	Single BERT model fine-tuned on multiple datasets to generalize across them. Achieved high accuracy but required extensive cross-corpora fine-tuning, sensitive to dataset differences.
Zouak et al. (2025) [7]	BERT + GraphSAGE (Graph Neural Network)	Enron, SpamAssassin, Ling	98.9–99.2% accuracy	Hybrid text-and-graph approach reached ~99% on standard corpora. Incorporated relationship graph features, though added complexity (graph construction overhead) and relied mainly on text content.
Roumeliotis et al. (2024) [18]	BERT vs. DNN, CNN, LSTM, CNN-LSTM (comparison)	Phishing/Spam email set	99.0–99.39% (BERT); 98.29–99.30% (GPT-4)	BERT outperformed classical deep networks (which were ~98% range). Noted the high computational cost of BERT and recommended compression/optimization for real-time deployment.

A. Synthesis of Model Performance & Adoption Trends

Over the last five years, the trends in performance and adoption of email spam detection have clearly shifted from relying on engineered features to utilising pretrained language understanding. Traditional machine-learning filters, particularly Naïve Bayes and Support Vector Machines, consistently achieve performance levels between 90% and 97% on widely used datasets, provided that features are thoughtfully designed and the distributions remain stable [1], [2]. Their strengths lie in speed and simplicity; however, the dependence on bag-of-words cues restricts semantic sensitivity and renders them vulnerable to obfuscation or shifts in vocabulary and style [2]. Deep neural networks have significantly advanced the field by directly learning phrase- and sequence-level structures from text. Convolutional Neural Networks (CNNs) and bidirectional Long Short-Term Memory networks (bi-LSTMs) consistently achieve over 98% accuracy on various benchmarks [3], [4], [23]. Convolutional Neural Networks (CNNs) effectively identify local patterns and variations in the spelling of spam lexicons without the need for manual engineering, thereby enhancing precision in the analysis of short to medium-length emails [3], [23]. Bi-LSTMs, on the other hand, are designed to capture longer dependencies and discourse cues, which enhance the ability to recall subtle phishing attempts and lengthy messages [4]. Nonetheless, deep models can exhibit variability across different datasets due to changes in topics, label distribution, and writing styles. This highlights the practical limitations of generalisation that are significant in real-world applications [1], [2]. The significant advancement comes with transformers: refined BERT-family models provide bidirectional context and comprehensive self-attention, consistently achieving approximately 98–99% accuracy with robust F1 scores across datasets such as Enron, SpamAssassin, Ling, and various mixed sets [6], [7], [25]. Pretraining provides a wide range of linguistic foundations, while task-specific fine-tuning allows for rapid adaptation with a limited amount of labelled email data, thereby reducing development cycles [16]. This combination has established transformers as the prevailing standard for text-only spam classification in recent research [6], [7], [25]. Another significant trend is efficiency: DistilBERT maintains a significant portion of BERT's accuracy while also decreasing latency and memory usage, which allows for more efficient service-level budgets in production environments [17]. Recent comparative studies indicate that, in addition to accuracy, factors such as throughput and inference cost are also being assessed. This shift highlights the importance of deployment considerations in email gateways and security appliances [18]. In contemporary architecture, numerous pipelines implement a lightweight transformer for initial triage, while reserving more intensive re-scoring for messages that are ambiguous. A two-tier approach helps keep latency low while still making good decisions [18]. Hybrid approaches are becoming the norm. For example, combining BERT with GraphSAGE enables a filter utilise both rich text context and graph signals (such shared senders, campaign links, and time) to show reputation and campaign structure [7].

Variants that incorporate feature selection cut down on noisy dimensions, which makes the boundaries tighter and the computation faster [12]. In parallel, explainable transformer methods and technology are being researched to make outputs auditable and boost analyst trust in security operations [15].

Many systems currently cluster around 99% accuracy on typical text-only benchmarks. This means that the focus is changing from getting small improvements to making models more robust and easier to transfer. This means being able to deal with drift, handle infrequent but high-impact assaults, and move cleanly between domains. [7], [1], [2]. Efficiency is just as important: to fulfil millisecond-level SLAs at the mailbox-provider scale, you require distillation, quantisation, and hardware-aware inference [17, 18]. Evaluation practice is also catching up, focussing on accuracy and recall in situations with very few false positives that show how much risk is acceptable in production [2], [10]. Transformers are being used in more than only English spam and phishing. This makes their coverage wider and their defences stronger against several types of attacks [11]. In general, the trend is towards small, pretrained transformers that can be improved with side signals and interpretability. This turns lab victories into filters for real-world email systems that are reliable, scalable, and can be audited [7], [17], [18].

B. *Research Gaps in Spam Classification*

1) *Real-time evaluation and efficiency:* Many research efforts continue to focus on achieving high static accuracy using carefully selected datasets, while there is often insufficient attention given to factors such as latency, memory usage, and sustained throughput in real-time scenarios [10], [18]. Integrating considerations of time and space complexity, along with streaming evaluations and stress tests—such as those assessing burst traffic and diurnal patterns—would provide a more comprehensive understanding of deployment readiness and the total cost of ownership [10].
2) *Explainability and auditorability:* Deep and transformer models that perform at a high level often lack transparency; there are limited studies that offer clear, interpretable explanations for decisions regarding spam or ham, which poses challenges for user trust and compliance efforts [10], [15]. Improvements in reliable attention visualisations, auditing of token importance, and clear rationales are essential to connect model accuracy with operational transparency in environments where security is a critical concern [10].
3) *Generalization across types and languages:* Many systems are English-centric and evaluated on a narrow set of corpora. Portability across languages, organizations, and spam modalities (e.g., phishing vs. promotions) remains under-explored; domain shift routinely degrades performance [1], [6], [10], [11]. Building multilingual, multi-category benchmarks—and methods that maintain performance across them—remains a priority
4) *Robustness to adversarial and evolving spam:* Attackers continuously introduce obfuscations (misspellings, image text, paraphrases) and novel lures. Robust training/evaluation against morphing and deliberately adversarial inputs is rare, despite evidence that evolving tactics reduce generalization [1], [10], [19]. Work on adversarial training, augmentation with realistic perturbations, and detectors for cloaking/URL manipulation is needed to sustain performance over time [10], [19].
5) *Hierarchical and fine-grained classification:* Most pipelines stop at spam/ham. There are few studies on hierarchical or multi-tier labeling that organize legitimate mail by topic/intent or distinguish phishing subtypes at scale, often due to annotation costs [5], [7], [4]. Zero-/few-shot methods and weak supervision could enable richer taxonomies without prohibitive labeling.
6) *Online learning and self-adaptation:* Models are typically static after training, yet spam drifts rapidly. Methods for safe online updates, human-in-the-loop feedback incorporation, and life-long learning remain underutilized, despite repeated calls for dynamic updating in reviews and position papers [1], [2], [9], [10].

Addressing these gaps would produce email classifiers that are not only accurate in the lab but also resilient, transparent, and broadly applicable across languages, organizations, and evolving threat surfaces.

C. *Current Challenges in Deployed Spam Filters*

1) *Computational cost and scalability:* Deep CNNs, full BERT, and LLMs impose heavy compute/memory footprints. At provider scale, even hundreds of milliseconds per message can be prohibitive; costs may exclude smaller organizations from state-of-the-art defenses [10], [17], [18]. Accordingly, pruning, quantization, and distillation—and hardware acceleration—are essential to maintain throughput at acceptable cost while preserving accuracy [10], [17], [18].
2) *Evolving spam tactics*: Zero-day phishing, high-quality spear-phishing, image-based spam, and AI-generated lures can evade text-only filters. Static models degrade within months without adaptation. Robustness and continual learning are therefore not optional in production; defenses must generalize to unseen attacks and refresh using recent evidence [1], [10], [19].

3) *Class imbalance and rare events:* After upstream filtering, ham often dominates inbox traffic; in that regime, even a 0.1% false-positive rate can be unacceptable. Conversely, rare but critical threats (e.g., CEO fraud) lack abundant training examples. Threshold calibration, cost-sensitive learning, cascaded designs, and careful validation under skew are mandatory to maintain user trust [10].
4) *Enterprise integration and operations:* Beyond algorithms, production filters must integrate with mail servers, directories, and quarantine/review workflows; they must be fault-tolerant, privacy-preserving, and auditable. Usability, legacy compatibility, and policy compliance frequently determine adoption success as much as accuracy [1], [9], [10]. Customization (tunable aggressiveness, organization-specific rules) further increases engineering complexity and maintenance burden.
5) *Maintaining near-zero false positives:* Users tolerate occasional spam leakage but not lost legitimate mail. Achieving extremely low false positives typically requires multi-layer pipelines (conservative primary filter, defer/flag strategies for uncertain cases) and, ideally, intelligible explanations so users can calibrate or override decisions. Opaque models make this balance harder to strike at scale [10], [1].

In production, accuracy is necessary but insufficient. Sustainable spam filtering hinges on efficiency (cost/latency), adaptability (online updates), reliability (fault tolerance), and trust (explainability, low false positives). Hybrid, hierarchical pipelines—pairing efficient transformers with targeted second-stage reasoning—offer a practical path forward, provided they are engineered with these operational constraints in mind [7], [18], [5].

IV. CONCLUSION AND FUTURE SCOPE

This review examined the progression of email spam detection from rule-based filters and classical machine learning to deep neural networks, transformers, and large language models (LLMs). Within this landscape, our hierarchical, two-stage architecture demonstrates that fine-tuning a compact transformer (DistilBERT) can deliver state-of-the-art spam filtering, achieving ~99.2% accuracy—surpassing a strong classical baseline such as a linear SVM (~98.3%). The second stage employs a zero-shot LLM committee (e.g., GPT-4, Gemini, DeepSeek-Chat) to assign semantic sub-categories to legitimate mail, reaching 91.6% accuracy with a macro-F1≈0.913 without any task-specific training. Majority-vote ensembling across LLMs consistently outperforms any single model, mitigating model-specific biases and stabilizing predictions. Together, these results show that high-accuracy spam detection and fine-grained ham organization can be achieved in one cohesive pipeline—moving beyond the binary framing that dominates prior work.

Practical viability follows from the way the pipeline allocates computation. Stage 1 leverages DistilBERT's contextual understanding and efficiency to remove the large majority of spam at low latency; Stage 2 applies deeper semantic reasoning only to the much smaller fraction of ham, enabling automated routing of alerts, support inquiries, updates, and similar categories. In mixed traffic, the overall system accuracy is ~95%, with near-perfect spam recall and LLM-committee latencies around 4–6 seconds per message—appropriate for asynchronous handling of ham while keeping the critical path fast. This division of labor makes the pipeline scalable and cost-aware, aligning well with enterprise needs such as triage, prioritization, and reduction of manual email handling.

Notwithstanding these gains, several constraints remain. First, explainability is limited: both transformer and LLM decisions are difficult to interpret, which complicates user trust, auditing, and compliance in regulated settings. Second, real-time efficiency at provider scale is non-trivial. Even with a lightweight transformer front-end, very large volumes and bursty traffic can expose throughput bottlenecks; careful optimization of inference, batching, and serving infrastructure is essential. Third, data limitations persist. Class imbalance across ham sub-categories can bias learning, and most publicly available, labeled corpora are English-centric, leaving performance in multilingual or domain-shifted environments under-validated. Fourth, adversarial evolution is continuous: new obfuscations, phishing templates, and content styles will erode performance unless models are refreshed. Finally, enterprise integration raises engineering and policy questions—data privacy (especially when cloud LLM APIs are involved), reliability, fallbacks, and interoperability with legacy systems. Addressing these facets will determine whether promising research prototypes translate into robust, maintainable production services.

A. *Future Scope*

1) *Multilingual and cross-domain generalization:* Extend training and evaluation to non-English corpora and adjacent channels (e.g., SMS, help-desk tickets). Building balanced, diverse datasets and using multilingual encoders or cross-lingual adaptation will improve robustness across geographies and organizational domains.

2) *Interpretable transformers and LLMs:* Incorporate explanation mechanisms directly into the pipeline (e.g., attention visualizations, token-level influence scores, example-based rationales, or concise natural-language justifications). Lightweight, human-readable evidence can increase trust, accelerate incident response, and support compliance.
3) *Adaptive and online learning:* Implement continual-learning loops that ingest fresh spam/ham signals (including user feedback) to counter concept drift. Safe, incremental updates—shadow deployment, canary testing, and rollback plans—can keep models current without disrupting service
4) *Expanded ham taxonomy and hierarchy:* Move from a handful of broad ham classes to a richer, hierarchical taxonomy (e.g., alerts → security vs. billing; promotions → transactional vs. marketing). Combine weak supervision, few-shot prompts, and active learning to scale labels with minimal annotation burden, while retaining high precision for user-facing automation.
5) *Performance and cost engineering:* Systematically benchmark end-to-end latency and throughput under production-like loads. Pursue quantization, pruning, distillation, and caching to minimize cost per message while preserving accuracy. Where privacy or cost precludes external APIs, evaluate distilled or optimized on-prem LLMs to keep inference local.
6) *Privacy-preserving deployment*: Explore confidential computing, in-house hosting, or hybrid architectures (local first stage; optional, privacy-screened second stage) to satisfy data-governance requirements. Clear data-handling policies and audit trails should accompany any LLM-assisted decisioning on email content.
7) *Defense against evolving tactics:* Harden training with attacks that look like the real thing: inject character mangling ("v1@gra"), template rewrites, and cloaked/redirected URLs, and add detectors that flag tell-tale signs of AI-authored text. Run periodic red-team drills to uncover blind spots and guide targeted fixes.

Stepping back, a layered pipeline works best: a compact transformer handles the fast path, while a small committee of LLMs weighs in only on the tricky cases. This split delivers strong spam catch rates, keeps legitimate mail organized, and remains practical to deploy. To make it production-ready at scale, prioritize four enablers: clearer explanations, broader language coverage, safe continual/adaptive learning, and cost-aware inference. In short, pair compressed transformers for routine screening with carefully governed LLM reasoning for edge cases—the combination yields a dependable, auditable, and user-centred spam filter.

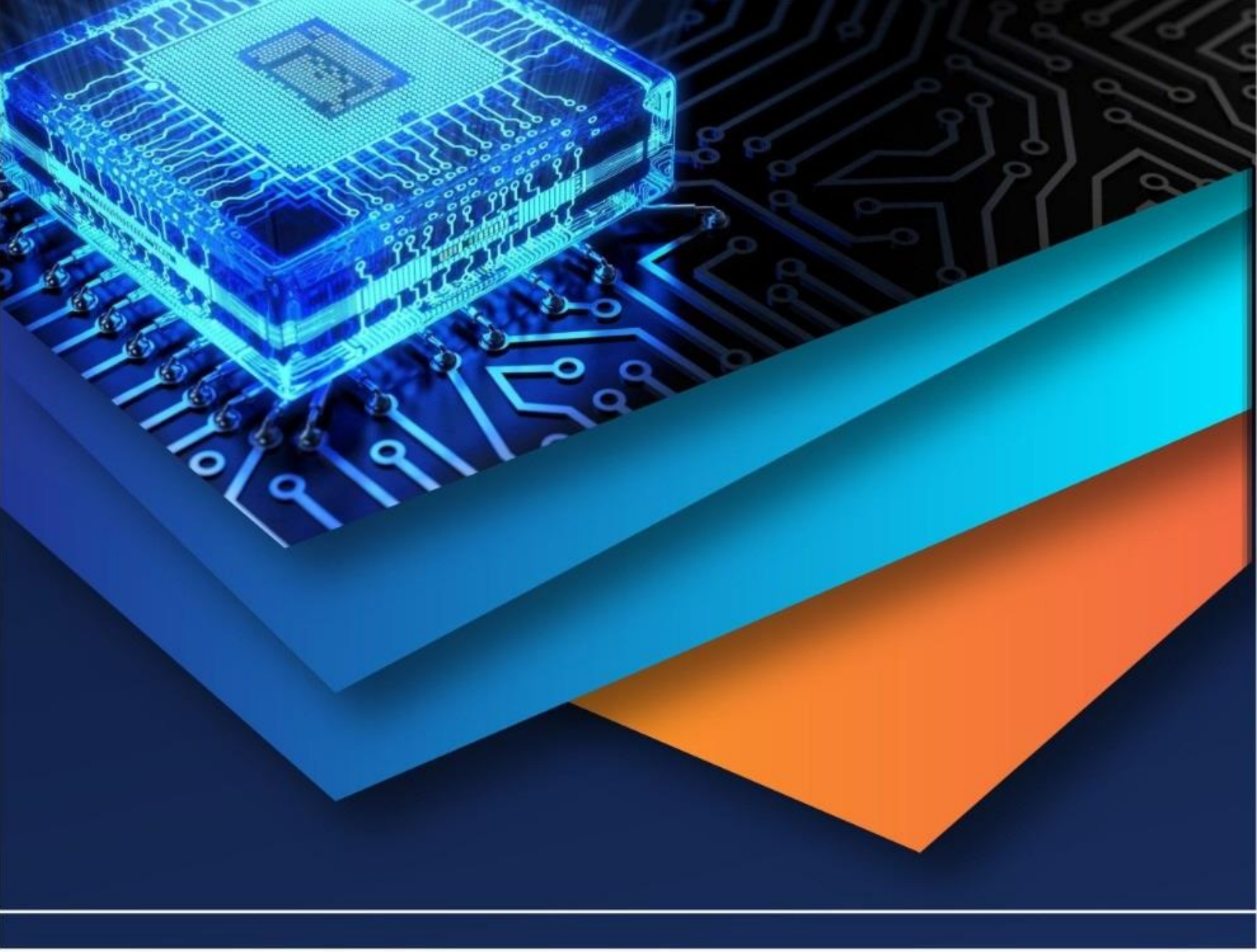
REFERENCES

[1] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "A review of spam email detection: analysis of spammer strategies and the dataset shift problem," *Artificial Intelligence Review*, vol. 56, no. 2, pp. 1145–1173, 2023. doi: 10.1007/s10462-022-10195-4.
[2] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, e01802, 2019. doi: 10.1016/j.heliyon.2019.e01802.
[3] A. Barushka and P. Hájek, "Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks," *Applied Intelligence*, vol. 48, no. 10, pp. 3538–3556, 2018. doi: 10.1007/s10489-018-1161-y.
[4] S. Zavrak and Ş. Yılmaz, "Email spam detection using hierarchical attention hybrid deep learning method," *Expert Systems with Applications*, vol. 233, 120977, 2023. doi: 10.1016/j.eswa.2023.120977.
[5] J. Doshi, K. Parmar, R. Sanghavi, and N. Shekokar, "A comprehensive dual-layer architecture for phishing and spam email detection," *Computers & Security*, vol. 133, 103378, 2023. doi: 10.1016/j.cose.2023.103378.
[6] V. S. Tida and C.-Y. Hsu, "Universal spam detection using transfer learning of BERT model," in *Proc. 55th Hawaii Int'l Conf. System Sciences (HICSS)*, 2022. doi: 10.24251/HICSS.2022.921.
[7] F. Zouak, O. El Beqqali, and J. Riffi, "BERT-GraphSAGE: hybrid approach to spam detection," *Journal of Big Data*, vol. 12, 128, 2025. doi: 10.1186/s40537-025-01176-9.
[8] Y. Wu, S. Si, Y. Zhang, J. Gu, and J. Wosik, "Evaluating the performance of ChatGPT for spam email detection," *arXiv:2402.15537*, 2024. doi: 10.48550/arXiv.2402.15537.
[9] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges," Security and Communication Networks, vol. 2022, Art. ID 1862888, 2022. doi: 10.1155/2022/1862888.
[10] E. H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi, and M. Uddin, "Email spam: a comprehensive review of optimized detection methods, challenges, and open research problems," IEEE Access, vol. 12, pp. 143627–143657, 2024. doi: 10.1109/ACCESS.2024.3467996.
[11] R. Meléndez, M. Ptaszyński, and F. Masui, "Comparative investigation of traditional machine-learning models and transformer models for phishing email detection," Electronics, vol. 13, no. 24, 4877, 2024. doi: 10.3390/electronics13244877.
[12] G. Nasreen, M. M. Khan, M. Younus, and B. Zafar, "Email spam detection by deep learning models using novel feature selection technique and BERT," Egyptian Informatics Journal, vol. 26, 100473, 2024. doi: 10.1016/j.eij.2024.100473.
[13] K. F. Rafat, M. F. Shahbaz, M. S. Memon, W. A. Khan, and F. Akhund, "Evading obscure communication from spam emails," PLoS One, vol. 17, no. 2, e0263451, 2022. doi: 10.1371/journal.pone.0263451.
[14] S. Saleem, Z. U. Islam, S. S. U. Hasan, H. Akbar, M. F. Khan, and S. A. Ibrar, "Spam email detection using long short-term memory and gated recurrent unit," Applied Sciences, vol. 15, no. 13, 7407, 2025. doi: 10.3390/app15137407.
[15] M. A. Uddin, M. Mahiuddin, and A. A. Chowdhury, "Explainable email spam detection: a transformer-based language modeling approach," in Proc. 27th Int. Conf. Computer and Information Technology (ICCIT), 2024, pp. 1–6. doi: 10.1109/ICCIT64611.2024.11022595.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186. doi: 10.48550/arXiv.1810.04805.
[17] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019. doi: 10.48550/arXiv.1910.01108.
[18] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Next-generation spam filtering: Comparative fine-tuning of LLMs, NLPs, and CNN models for email spam classification," Electronics, vol. 13, no. 11, 2034, 2024. doi: 10.3390/electronics13112034.
[19] M. Schmitt, P. M. Sowa, K. Huguenin, B. Smeets, and P. Simoens, "Digital deception: generative artificial intelligence in social engineering," Artificial Intelligence Review, 2024. doi: 10.1007/s10462-024-10973-2.
[20] M. Stow and B. S. Ezonfa, "Two-stage email classification model for enhanced spam filtering through feature transformation and iterative learning," International Journal of Computer Sciences and Engineering, vol. 13, no. 2, pp. 16–27, 2025. doi: 10.26438/ijcse/v13i2.1627.
[21] H. A. Al-Kabbi, M.-R. Feizi-Derakhshi, and S. Pashazadeh, "A hierarchical two-level feature fusion approach for SMS spam filtering," Intelligent Automation & Soft Computing, vol. 39, no. 4, pp. 665–682, 2024. doi: 10.32604/iasc.2024.050452.
[22] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," IEEE Access, vol. 7, pp. 168261–168295, 2019. doi: 10.1109/ACCESS.2019.2954791.
[23] A. Poobalan, K. Ganapriya, K. Kalaivani, and K. T. Parthiban, "A novel and secured email classification using deep neural network with bidirectional long short-term memory," Computer Speech & Language, vol. 89, p. 101667, 2024. doi: 10.1016/j.csl.2024.101667.
[24] M. Adnan, M. O. Imam, M. F. Javed, and I. Murtza, "Improving spam email classification accuracy using ensemble techniques: a stacking approach," International Journal of Information Security, vol. 23, no. 1, pp. 505–518, 2024. doi: 10.1007/s10207-023-00756-1.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)