ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/rahul-kumar/622

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 3 & Iss: 4

Research Article

A Comparative Analysis of Machine Learning and Deep Learning Architectures for Spam Mail Classification

Rahul Kumar* and Gaurav Garg

Department of Computer Science and Engineering, University Institute of Engineering, Chandigarh University, Mohali-140413, Punjab, India

Citation: Kumar R, Garg G. A Comparative Analysis of Machine Learning and Deep Learning Architectures for Spam Mail Classification. *J Artif Intell Mach Learn & Data Sci 2025* 3(4), 3000-3008. DOI: doi.org/10.51219/JAIMLD/rahul-kumar/622

Received: 10 November, 2025; **Accepted:** 15 November, 2025; **Published:** 17 November, 2025

***Corresponding author:** Rahul Kumar, Department of Computer Science and Engineering, University Institute of Engineering, Chandigarh University, Mohali-140413, Punjab, India, E-mail: rahulkumar8051rt@gmail.com

ABSTRACT

Email spam remains one of the most persistent and insidious cyber threats, accounting for a significant portion of utilized network resources, lowering users' productivity and standing as a well-known threat vector for malicious activity like phishing and malware distribution. The paper offers a comparative review of spam mail classification methodologies, focusing on the transition from traditional machine learning to deep learning architectures. We analyze the performance of basic and advanced algorithms, such as Naive Bayes, SVM and Random Forest, against Convolutional Neural Network, Long Short-Term Memory networks, state-of-the-art transformer models like BERT. The analysis is supplemented by a systematic review of existing benchmarks based on several traditional datasets, namely UCI Spambase and data corpora originating from Enron and SpamAssasin. Our major finding argues that while classical ML approaches demonstrate excellent performance efficiency and accuracy on pre-customized feature sets, deep learning models, especially BERT, demonstrate superior performance on text-based datasets due to the awareness of deep context. The research reveals a crucial trade-off between classification accuracy and computational complexity. Finally, the paper concludes with the discussion of current challenges and asserts that the future of spam filtering corresponds to the development of adaptive hybrid architectures.

Keywords: Spam Classification, Machine Learning, Deep Learning, Natural Language Processing, Naive Bayes, Support Vector Machine, LSTM, BERT, Text Classification

1. Introduction

Email has become an essential tool of common communication in almost any professional or personal setting, be it business, homeschooling or research. Nevertheless, its accessibility factor has always been put under the dark light of another kind of complete of unsolicited and irrelevant bulk email, also known as spam. Contrary to light threats, spam is an actual threat. It causes the overload of network traffic, utilization of scarce storage capacity and computational capacity and significant loss of user time and productivity. Furthermore, it is the most specific way of delivering major security threats, such as phishing attempts for sensitive data theft and malware and ransomware distribution. The flood of spam email has risen exponentially, challenging both email service providers and their customers.

The fight against spammers has ultimately led to a technological "arms race" with filter developers. Spammers create new evasion schemes as ingenuity of protection methods increases. While some of these are simpler and involve hiding

harmful content in seemingly innocent text or changing sender identification, others require more sophisticated techniques, such as inserting spam into image files to avoid recognizing it through text search. The adversarial and evolving nature of the problem means that the filtering solutions must become increasingly adaptive, intelligent and robust to keep up with new alleviation techniques.

The development of technology from static rule-based methods to MLPS technologies illustrates these trends and the evolution of new threats. Initial solutions in the 1990s relied on simple keyword analysis and manual curation of filters, e.g., mailing lists. These were static approaches that were not able to cope with new alleviation or users' individual data and often caused errors. The introduction of statistical methods, particularly Bayesian filtering, was the first major breakthrough in transforming systems into learning-based.

This paper tabulates a comprehensive, data-driven comparative analysis of the modern spam classification methodologies. The unique contribution is the methodical evaluation of fundamental machine learning models vis-à-vis cutting-edge deep learning architectures. Through synthesizing the relevant performance benchmarks in the literature on multiple standard datasets, this paper seeks to chart out the relative strengths, weaknesses and optimal use-cases of each class of models. The remainder of the paper is structured as follows: Section II provides a detailed review of the development of spam filtering literature; Section III introduces the experimental protocol, including the standard datasets, preprocessing protocols and evaluation metrics employed in this comparative analysis; Section IV describes the performance results and conducts a comparative analysis across the sets of models on the diverse dataset; Section V discusses the broader implications of the results in light of the current challenges and opens areas of research. Finally, Section VI concludes the paper by summarizing the main findings.

2. The Evolution of Spam Filtering Techniques: A Review

It is evident that the development of spam filtering technologies has evolved from static and simplistic rules to machine learning systems that can analyze vast volumes of data. Such a trajectory has been predetermined by the evolution of spamming and is consistent with the overall development of the artificial intelligence and the NLP field.

A. Early approaches: Rule-based and statistical filtering

The rapid rise in the amount of unsolicited email to be blocked in the latter half of the 1990s gave rise to the first generation of spam filters. These first systems were largely rule-based at the time and operated on pattern-matching straightforward systems.

- **Keyword and Rule-based systems:** They typically utilized a set of rules and keyword lists to assess an email. If an email included certain words or phrases frequently found in spam, including free, win, guaranteed or limited time offer, it would be considered spam. Leading open-source projects such as Spam Assassin, for example, started with a big number of such rules, which tested for specific keywords, information header anomalies and the sender's reputation depending on blacklists. In several aspects, while good at catching the most apparent spam, these systems had significant disadvantages including being rule-based were inflexible and attempted to keep up with the constantly changing spam tactics. Additionally, a high rate of inaccuracy, the phenomenon of legitimate emails being mistakenly labelled as spam, was infamous.
- **Bayesian filtering:** The turning point in the use of spam filters was the emergence and popularization of Bayesian filtering in the early 2000s. For the first time, the use of spam filters did not imply a set of static rules but statistics-based training. The principle of Bayesian filter is as follows: the filter analyses the content of the email received and calculates the probability that it is spam based on the frequency of the words inside it for spam and proper emails. Moreover, based on user feedback, for instance, how users have marked emails as "spam" or "not spam". Therefore, over time, the filter became more personal and infected individual unique characteristics or patterns of the e-mail usage. Such filters made it possible to significantly improve accuracy, reduce the number of false positives and for the first time, consider the most important – they form the basis of training systems that are uncommon today.

B. Foundational machine learning classifiers

The success of Bayesian filtering set the stage for the utilization of more formal machine-learning methods, which cast spam detection as a supervised binary classification problem. More precisely, a model is trained on a big amount of labeled emails to determine a decision boundary that can be utilized to classify new, unseen emails. Those models are usually trained using the following ML pipeline: data preprocessing, feature extraction, model-building and performance assessment. This branching of algorithms was highly successful:

- **Naive Bayes (NB):** A direct progression of Bayesian principles, the Naive Bayes classifier is a probabilistic model grounded on Bayes theorem. It establishes the likelihood of an email being a member of a class given a set of features and then compares these probabilities, choosing the class with the highest likelihood. It is "naive" because of its assumption – all features are conditionally independent given the class while it is not true for language, it makes the computation simple and yielded great results. NB remains a strong baseline due to its simplicity and low false positive rate.
- **Support Vector Machines (SVM):** SVM algorithm is a strong classifier. SVM operates by determining the optimal hyperplane that widely separates the data points of different classes in a high-dimensional feature space. SVMs are more effective in classifying large feature spaces with text classification. For a large vocabulary set, a model-based kernel regression model can be implemented to increase the classification capacity.
- **Decision Trees (DT) and Random Forests (RF):** The decision tree is an intuitive tree-structure model such that internal nodes include features, branches include decision rules and leaf nodes represent class labels. The main advantages are explained are the interpretability and understanding of the decision-making process. In practice, it may overfit the training data if one utilizes only a single decision. Random Forest that is the ensemble learning method that constructs multiple decision trees during training and outputting the class that is the mode of the

classes of the individual trees is used to overcome this disadvantage. Random forest reduces variance, improves accuracy and generalizes the model that is more robust against overfitting, owing to averaging the predictions of many trees.

C. The advent of deep learning in text classification

Another major development was the evolution from traditional ML to deep learning, which was made possible by the capacity of deep neural networks to automatically train multilayered and multistage feature representations straight from raw data. Specifically, the aforementioned technological breakthrough diverged from the traditional ML pipelines, with their deserved focus on labour-intensive manual feature engineering.

- **Recurrent architectures (RNN, LSTM, GRU):** Recurrent Neural Networks are specially tailored to work with sequential data such as text. However, "simple" versions of RNNs cannot be used for long-range dependencies in a sequence. Long Short-Term Memory and Gated Recurrent Unit networks were developed to solve this issue – the so-called "gate" mechanisms allow the network to "decide" what to remember and what to forget over long sequences. The ability to retain long-term context is critical for understanding most aspects of the language. Bidirectional variants such as Bi-LSTM also became popular nowadays – unlike the "simple" architecture, Bi-LSTM processes text in two directions at once and produces "two views" of the significance of word indices in the context.
- **Convolutional neural networks (CNNs) for text:** Indeed, although CNNs are most popular for computer vision, they can be re-purposed for text classification. A 1-D convolutional layer applies filters that slide over the input word embedding sequences and acts as a bank of n-gram detectors, capturing local patterns and word sequences. Filters of varying sizes catch important linguistic patterns of different lengths such as trigrams, 4-grams and so on that are indicative of spam. Therefore, CNN is an efficient and effective method for text classification.

D. State-of-the-art: Transformer architectures

The most recent revolution in NLP and text classification is the Transformer architecture, which has achieved state-of-the-art performance across a variety of tasks.

- **The attention mechanism:** The key breakthrough in Transformers, as the name suggests, is the self-attention mechanism.7 Instead of reading the input one word at a time, as RNNs do, self-attention allows the model to assign a weight to every other word in the input sequence for each word in the sequence. This allows the model to create an extremely contextualized representation of each individual word, as it directly models the relationship between every pair of words in the text, regardless of distance from one another.
- **BERT (Bidirectional Encoder Representations from Transformers):** A seminal model that epitomizes the Transformer architecture is BERT. It is pretrained on a colossal corpus of text with unsupervised objectives, such as Wikipedia and Books Corpus, to acquire a deep understanding of language structure and meaning that is bidirectional. Thereafter, the pre-trained model can be directly fine-tuned to specific tasks by training the model on a smaller, labelled dataset that is task-specific, for instance, a dataset for classification of spam.

The method has been all the rage since its inception, with BERT and its variants and their variants DistilBERT, RoBERTa currently leading to new state-of-the-art results in spam detection by exploiting subtle contextual cues that older models would otherwise overlook.

The common thread in this technological journey is an evolution in the understanding of spam and the methods for detecting it. The transition is not just a matter of gradual model complication but instead represents a more critical transformation in the analytical approach. First, systems based on rule used pattern-matching software, looking for particular words and patterns. Systems became more sophisticated and adopted machine learning, moving to rule-based statistical organizations such as Naive Bayes, support vector machines, etc. Second, in the progression scheme, rule-based systems are replaced by statistical learning systems, which do not really know what is going on in any text. Instead, the system uses pairs of features and classes and their statistical relationships without understanding the meaning or how the pairs are connected, it excels at identifying spam. In the last progression step, based on deep learning and especially Transformers, move to understanding the context. It does not just track words, measures them and counts them but also knows that some words depend on others.

This technological advancement is a response to the fact that spammers evolve simultaneously but differ from simple keyword spammers to others who design more sophisticated texts. These texts pass the linguistic analysis that models using SVM and Naive Bayes have no chance to pass.

3. Experimental Framework for Comparative Analysis

In order to have a fair and meaningful comparison of these spam classification approaches, it is important that a consensus on experimental setup be achieved. This involves the definition of benchmark datasets, description of the data preprocessing and feature extraction procedures as well as specifying model architectures and evaluation metrics. This section presents the reconstituted methodological approach based on a literature review.

A. Benchmark datasets: Characteristics and composition

The accuracy of any classification model depends on the data that is used for training and testing. In case of spam classification, the space of high-volume public datasets is limited to a few canonical ones that enable testing varying dimensions of models' abilities.

- **UCI spambase:** It is one of the most commonly used data set in ML for spam labelling. 4601 instances - processed e-mails from a single folder (file 'inmail.mbox') Each instance corresponds to one e-mail and it is described by 57 features + the class. 28 These features consist of the counts of 48 specific words (e.g.,"make", "free"," credit"), as well as the counts for 6 'characters'(e. g., '!', '$') and\" `three metrics comparing sequences of capital letters. 28 Note that the raw text of the emails is not available in this

dataset. around 39.4% labeled Sp or spam, the remaining as legitimate (ham). 28 Its pre-speced, numeric based content type serves as a good benchmark data source for testing conventional ML algo- rithms without NLP overhead.

- **Enron-spam:** The Enron Corpus is one of the largest publicly available mass collections of "real" emails (spanning a variety of uses and purposes as well as types and qualities) and is therefore invaluable for realistic evaluation. 12 The Enron-Spam dataset is a part of the corpus, built for spam filtering studies. A popular one contains 33,716 emails (half spam ${\rm ham}=16,545, {\rm spam}=17,171$). 34 Unlike UCI Spam base, it contains raw email text (including headers, subject line and body of the message). 34 This unstructured nature of data requires end-to-end NLP preprocessing pipeline and serves as a challenging testbed for deep learning models that are designed to learn from raw text.
- **SpamAssassin:** Another large set of plain text messages is the SpamAssassin public corpus. One variant of this corpus is composed of 6,047 messages out of which about 31% are considered as spam. One interesting feature of the SpamAssassin corpus is that it separates natural mail into easy_ham (mail pieces should be easily differentiated from spam) and hard_ham (natural mail that may exhibit spam-like characteristics like HTML text formatting or advertising-speak). The hard_ham in this dataset makes it especially challenging as it probes a classification model's capability to recognize nuances in context and not trigger false positives. **(Table 1)** provides a summary of these key benchmark datasets.

Table 1: Characteristics of Benchmark Datasets.

Dataset Name	Number of Instances	Feature Type	Spam Percentage	Ham Percentage	Key Characteristics
UCI Spambase	4,601	57 pre-engineered numerical features	39.40%	60.60%	Structured data; no raw text; tests classifiers on statistical features[28,29].
Enron-Spam	33,716	Raw email text (subject, body, headers)	50.90%	49.10%	Large-scale, real-world data; requires full NLP preprocessing[12,34].
SpamAssassin	~6,000	Raw email text	~31%	~69%	Contains easy ham and hard ham; tests robustness against ambiguous cases[12,37].

B. Data preprocessing and feature engineering pipeline

The way to prepare data that will go into the models depends upon the shapes of these datasets.

Pre-processing of Raw Text (Enron, Spam Assassin):

In the case of unstructured text datasets, a conventional NLP preprocessing pipeline is used to clean and normalize the data for feature extraction. This process commonly involves the following phases:

- **Text cleaning:** Delete unwanted substances like html tags, special characters, punctuation and digits.
- **Normalisation:** Transcriptions will have case normalised - usually to lower and we treat all of 'Free' and 'free' as the same word.
- **Tokenization:** Taking the cleaned-up text and splitting it into a sequence of words or pieces, called tokens.
- **Stop word removal:** It is the removal of high-frequency, but low information words (such as "a", "the", "in") that do not contribute to classification.

Lemmatization or Stemming: It is the process of linking words together i.e. A group of words stemming from the same root word (e.g running, ran and runs reduce to run). Lemmatization is often better as it takes the context of words to make a valid dictionary word, vs stemming which just looks at a plain list and tries to strip affixes.

Feature representation (Vectorization): After preprocessing, the text will need to be turned into numbers (vectors) that machine learning models can process.

TF-IDF (Term Frequency-Inverse Document Frequency): This is the classic vectorization approach for classic ML models. It generates a vector for each document, where the values of the dimensions are words found in the vocabulary. The value in each dimension is the TF-IDF (term frequency-inverse document frequency) score that indicates the importance of a word to a document in a corpus. The score grows with the frequency of the word in a document but is inversely proportional to its probability across the corpus, which allows common words like "the" (which occurs in many documents) to be penalized.

Word Embeddings (Word2Vec, GloVe): In deep learning models' dense vectors are typically preferable. Word embedding models such as Word2Vec and GloVe generate continuous vector presentations of words by learning from big corpora. These vectors reside in a multi-dimensional space and words with similar meanings end up close to each other also capturing the semantic meaning.

Contextual Embeddings (BERT): Current Transformer models such as BERT produce contextual embeddings. Unlike Word2Vec static embeddings, the word's vectorial representation is built on-the-fly according to its specific context. This enables the model to address word sense disambiguation (e.g., "bank" of a river v.s. financial "bank") and gather significantly richer semantic context.

C. Evaluated model architectures and implementation details

The models considered in this comparative study are the ones that were widely reported in the literature (Section II). For classical ML techniques, such as SVM many popular formulations use kernels, e.g., linear or Radial Basis Function (RBF). Common structures for deep-learning models are stacked multiple LSTMs or CNNs with dense layers after them for classification. Transformers using BERT as a fine-tuning for solving such tasks can be described by pretraining on the task-agnostic task and then simply classifier on top of the output at its end encoder.

D. Performance evaluation metrics

In order to quantify and compare classifiers performance, common metrics based on the confusion matrix have been used. The confusion matrix shows a more skintight image of how accurate the predictions of a model are compared to the true labels.

The Confusion Matrix: The confusion matrix is at the root most classification metrics and has four critical values.

- **True Positives (TP):** The number of spam emails that are correctly classified as spam.
- **True Negatives (TN):** The number of ham emails that are properly classified as ham.
- **False Positives (FP):** Number of ham emails that were misclassified as spam(emails) (i.e., Type I error).
- **False Negatives (FN):** The number of spam emails that were mistakenly deemed as ham (Type II error).
- **Standard Metrics:** The following metrics are calculated from these values.
- **Accuracy:** This shows the percentage of positive classifications made were actual positive. Although natural, this is a misleading measurement under class imbalance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** Indicates the ratio of predicted spam emails that were actual spam. Spam filtering has to have high precision to avoid too many legitimate emails being sent to a spam folder.

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

- **Recall (Sensitivity):** The ratio of all spam email to be identified by the filter.

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

- **F1-Score:** The harmonic mean of precision and recall, giving a single score that balances both. It is very useful when the class distribution is imbalanced and importance of both FP and FN difference.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For spam filtering the cost of false positives is usually regarded as being much higher than this for false negatives. Learning that important business or personal emails are marked spam can be quite harrowing as opposed to inclusive of that spam email which we so disdain. Thus, Precision will generally be the most important metric to consider in the real-world application of spam filters.

4. Results and Analysis

This section provides a consolidated review of the machine learning and deep learning models in terms of performance and computing complexity compiled from benchmark findings in the literature. The results are presented based on the unique nature of the three principal datasets, demonstrating how data complexity and structure affect model performance.

A. Performance on structured feature data: UCI spambase dataset

The UCI Spambase with numerical features pre-engineered is a good choice as a benchmark to evaluate the performance of conventional machine learning methods working on structured feature space. Here, the goal is not understanding of language, but to identify patterns out of these statistical figures we are given.

Indeed, in the literature it is well documented that many machine learning classifiers can obtain high accuracy on this data. Bagging-based ensemble methods, like Random Forests, are often one of the best classifiers with published accuracy rates commonly exceeding 95% and sometimes even as high as 99%. Similarly Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) also exhibit a hearty responsiveness with the accuracies in the bracket of 95-98%. Fast for the naive Bayes are very fast in computation, but they typically also have moderate (even if eventually slightly) reduced accuracy, e.g. often at 89-93% depending on the setting (though it can be competitive w.r.t. to precision and F1 when configured well in comparison to other models). The strong performance of these models further highlights that they are effective when given well-chosen and informative features. Table II summaries these findings and presents a comparative analysis of common performance measures for basic ML classifiers on the UCI Spambase dataset **(Table 2)**.

Table 2: Performance of Machine Learning Classifiers on the UCI Spambase Dataset.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	89.2 - 93.0	88.1 - 98.0	90.3	89.2
Support Vector Machine (SVM)	94.7 - 98.0	93.5	94.9	94.2
Random Forest (RF)	95.0 - 99.0	96.0 - 98.2	96.5 - 97.2	96.6 - 97.4
Decision Tree (DT)	91.3 - 92.0	~91.0%	~91.5%	~91.2%
Logistic Regression	94.3 - 96.0	~95.2%	~95.8%	~95.5%
Artificial Neural Network (ANN)	94.1 - 98.1	~97.0%	~97.0%	~97.0%

B. Results on unstructured text data: Enron-spam and spamassassin collections

The Enron-Spam data and SpamAssassin corpus, raw textual emails, pose a more difficult challenge as models have to simultaneously identify features in the text while capturing linguistic and contextual information. These are the datasets on which deep learning (DL) methods show they clearly outperform classical machine learning (ML) algorithms, essentially thanks to their automatic feature learning and contextual representation abilities.

Enron-spam results:

Deep learning models outperform the traditional Machine Learning based models on Enron dataset. Recently, transformer-

based models, such as those based on BERT (Bidirectional Encoder Representations from Transformers) have established state-of-the-art performances with reported accuracies in the 97–99% range and some recall rates even exceeding 99%. This is evidencing high sensitivity in spam detection and low to zero missed spam emails. The performance of recurrent networks such as LSTM and hybrid models (e.g., CNN+LSTM) are also competitive, as they attain accuracies consistently in the mid-to-high 90s. Although RF still is one of the best performing traditional ML algorithms on this dataset, in terms of accuracy it usually falls short to top-performing DL models with several percentage points; mainly due to its poor representational power (i.e., limited capacity) in capturing salient semantic and syntactic dependencies within natural language.

SpamAssassin results:

The same trends are shown in the SpamAssassin corpus, with more complicated extensions due to introduction of hard_ham category. And because it has borderline legitimate messages like promotional emails and HTML-ridden content, the classifier must learn to filter these out based on more subtle context cues or suffer a sharpening of the edge in both directions. As usual, also in case of deep learning and hybrid models achieve better performance; reported accuracies frequently surpass 98% by employing complex architectures. Leveraging pre-trained language models such as BERT or RoBERTa provides a great benefit as these models have encoded massive amounts of contextual language understanding that helps separate out legitimate but "spam-like" content from actual spam.

Beyond mere accuracy, these models excel in recall, precision and F1 score as well implying not just higher degree of detection but also less misclassification between the actual spam vs legitimate emails. In addition, as existing researches have demonstrated that extra domain-specific corpuses fine-tuning transformer models (e.g., fitting BERT to email communication style) possibly boost the generalization capacity.

Overall comparative insights:

Comparison of machine learning and deep learning models the comparative results between machine learning and deep learning on the Enron and SpamAssassin datasets are presented in **(Tables 3,4)**. The continued success of deep learning architectures and particularly transformer-based models demonstrates their power in automatic feature extraction, semantic understanding and contextual reasoning which classical ML models – depending on developing hand-engineered features - inherently do not have. These taken together confirm that deep learning is relatively more favorable as dataset complexity becomes larger with increasing distance from structured-representative features (UCI Spambase) to unstructured-coarse raw text (Enron and SpamAssassin).

Table 3: Comparative Performance on the Enron-Spam Dataset.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	87.5 - 92.8	~94.0	~90.2	~92.1
Support Vector Machine (SVM)	93.9 - 95.5	~93.0	~90.2	~95.0
Random Forest (RF)	95.5 - 98.4	~91.0	~92.0	~91.5
CNN	~86.0	~80.0	~93.0	~86.0
LSTM	94.9 - 98.5	~96.0	~96.0	~96.0
BERT	97.0 - 98.9	96.0 - 97.0	97.0 - 99.0	96.1 - 99.0

Table 4: Comparative Performance on the SpamAssassin Dataset.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	~94.6	~93.3	~94.7	~94.0
Support Vector Machine (SVM)	~95.2	~94.5	~95.1	~94.8
Random Forest (RF)	94.2 - 96.8	~95.7	~96.7	~96.2
CNN-LSTM Hybrid	~98.4	~98.0	~98.0	~98.0
Deep RNN	~99.7	~99.7	~99.7	~99.7
BERT	98.0 - 98.9	98.7	98.5	~98.6

C. Cross-paradigm comparison and visual analysis

We find that a visual comparison is very effective at combining the individual data sets.

(Figure 1) Bar Plot of Peak Accuracy in all Datasets. A bar chart would be built to illustrate the highest-reported accuracy of top-performing traditional ML model (i.e. Random Forest -RF-) and top-performing DL model (i.e., BERT) on each dataset. The too 90s UCI Spambase kind of performance bar would have similar height bars on RF and one of the best DL models. But for the Enron-Spam and SpamAssassin charts there would have been a visible separation, with BERT's bar much higher than Random Forest. This visualization would be a dramatic representation of the main headline: ML models fare very well on structured text, but DL models and especially Transformers, have an accuracy edge for unstructured text.

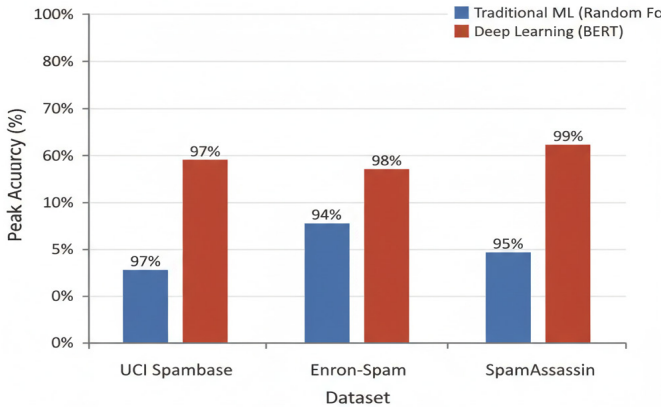

Figure 1: Bar Plot of Peak Accuracy in all Datasets.

(Figure 2) ROC Curves for Key Classifiers. A comparative ROC curve for Naive Bayes, SVM, Random Forest and BERT on the Enron-Spam dataset would also help in specifying performance distinction. The optimal curve for BERT should ideally be closest to the top-left corner, since it would indicate a better balance between obtaining high True Positive Rate (Recall) and low False Positive Rate over all possible classification thresholds. The AUC of the BERT would be the highest, probably to around 0.99, giving quantitative evidence in Favor of its better discriminative power as compared to that from other models.

D. Analysis of computational complexity and performance trade-offs

The higher accuracy of deep learning models, particularly large pre-trained models such as BERT, is accompanied by a considerable computational overhead. The process of training and finetuning these models is computationally expensive, demanding powerful GPUs and long hours. Unlike, traditional

ML approaches such as Naive Bayes are fast to train and computationally lightweight and can be employed in real-time or memory-constrained systems. The model SVM and Random Forest are in the middle of these two approaches. This trade-off between prediction performance and computational efficiency is an important issue for realworld deployment. A system designer may opt for a slightly less accurate but much faster model, either as a first pass filter or for running on mobile devices and apply the more computationally expensive model to analysing more dubious emails server side.

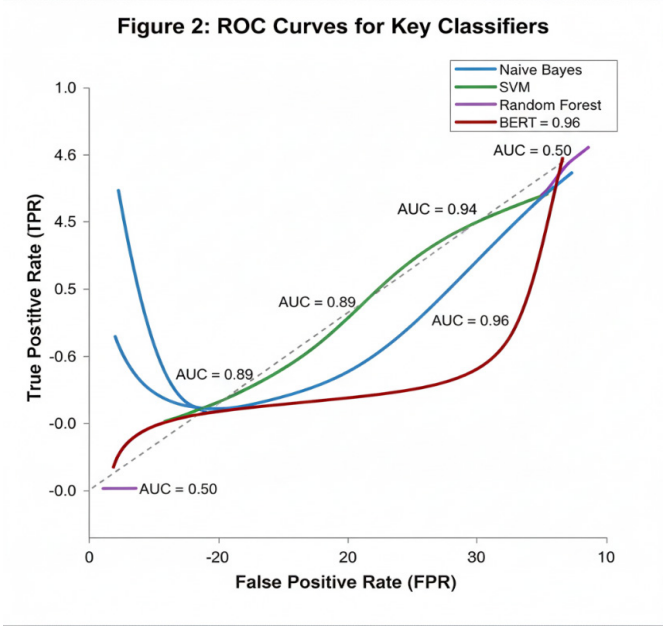

Figure 2: ROC Curves for Key Classifiers.

5. Discussion

The results reported in the previous section clearly compare different spam classification schemes quantitatively. We interpret these results, as well as other implications and identify new trends and future directions of spam detection in this section.

A. Identifying the strengths and weaknesses of different model families

The study shows the complexity associated with each family of algorithms has a specific profile where they excel and a threshold in the noise level, at which the performance decays.

- **Traditional machine learning (NB, SVM, RF):** This category is well-known for being efficient and high performing in structured or manually-created feature varieties. For example, with the UCI Spambase dataset, models such as Random Forests (in combination with preprocessing) are capable to obtain classification performances that can be compared with more elaborate architectures assuming the data is already fed in form of meaningful numerical features. Decision Trees especially provide some level of interpretability (which is pretty much non-existent with deep learning models) as they can be easily interpreted in terms of decision rules. The main limitation is that they are heavily dependent on manual feature engineering and can't learn deep semantic information from raw text due to the explicit features used (features in Fig 2), which reduces their ability to capture nuances or adversarial messages.
- **Deep learning (CNN/LSTM):** The defining benefit of this group is its feature learning capability from raw texts by itself. Models such as LSTMs are well-suited to capture sequential dependencies, whereas CNNs are effective in learning local patterns (n-grams). They are a powerful extension of traditional ML for learning structure in language. But Transformer architectures, while more expensive computationally, have largely outperformed them.
- **Transformers (BERT):** One key strength of Transformer-based models is their deep, pre-trained contextual language knowledge. BERT exploits information from a large corpus of text, making it very good at distinguishing subtle semantic cues and therefore being particularly resistant to spam that emulates authentic communication. This is their main strength and why they perform state-of-the-art. The most important drawback of such pre-trained models is the enormous amount of computational resources that are necessary for both training and fine-tuning these models, rendering them impractical for deployment in certain settings.

B. The impact of dataset characteristics on model performance

One important lesson to be learned from this analysis is that there is no "one best model" at all times, the performance of a model is very much dependent on the actual characteristics of the dataset.

The discrepancy between the performance of classic ML and deep learning (DL) models becomes larger with more difficult and ambiguous data. When dealing with the well-structured UCI Spambase dataset, the accuracy aggression of Random Forest (96–99%) compared with a complex ANN model (~98%) is negligible. This largely stems from feature processing, where patterns are extracted from the data during dataset construction and models just need to learn and generalize these patterns using well-defined numeric features.

Interesting gap can also be observed on raw-text collections (Enron-Spam and SpamAssassin). Even a highly fine-tuned RandomForest may cap out around 95% accuracy; whereas models based on transformers (like BERT) consistently boost performance to the 98-99% range. The SpamAssassin corpus, containing the hard_ham class of legitimate but "spam-like" examples, is a good example of this - these samples require finer knowledge of the stylistic content in order to be classified correctly. Exactly in such cases, the capacity of transformer-based models like BERT to understand both context and semantics make it scores over traditional ML algorithms that rely more on statistical pattern matching.

Overall, the results obviously indicate that when a dataset gets more realistic compared to real-world email traffic, deep learning methods are no longer only beneficial but imperative in order to accomplish consistent and universally applicable spam detection performance.

C. Emerging challenges and future research directions

Even though we have made a significant advancement in detecting spam, the area continues to be a fertile ground for spam. The problem is made intractable by the nature of hostility and this indomitable "arms race".

- **Concept drift:** Spam behavior is not constant, but rather

changes over time ¼uctuates to avoid being caught. The model that was trained on last year's data might not work for today's spam. This issue is known as concept drift and represents a significant challenge. In future work we will have to address this issue by concentrating our development towards the realization of an entirely unsupervised AL system capable of receiving data at any moment and being used to incorporate new spam patterns in real time avoiding full retraining from scratch.

- **Adversarial attacks:** Effectively everything you use automates the process of blocking spam, this also has a tendency to teach how machine learning classifiers work. Spammers are at this point actively creating e-mails designed specifically with an intent on fooling ML classifiers. This might include injecting benign terms in the spam message, using synonyms of commonly spammed words or obfuscating text by adding it to an image. Adversarialy-robust models are precisely the solution to this problem and researching them vigorously is paramount.
- **LLM-generated spam:** One of the most recent and threatening attacks is the generation of spam using Large Language Models (LLMs). These models can generate very coherent, syntactically and semantically correct and contextually relevant text, that is almost indistinguishable from human written. This new wave of AI-generated spam could easily slip past filters trained on older, more formulaic examples of spam. This will probably be the new next great frontier for spam detection techniques with a need to look more closely at recognizing statistical tell-tail signatures of machine-generated text.

These challenges imply that the future toward spam detection is not about discovering a unique, static "best model." It rather indicates the way towards dynamic, hybrid, adaptive systems. A pragmatic, cost-effective approach would be to use a multi-stage screening process: Some simple model like Naive Bayes could do a quick-and-dirty job of blocking the obviously bad emails; whereas scaling up to more complex/expensive Transformer-based models might make sense for analyzing marginally suspicious emails in more depth. Critically, such a system should include elements for incremental learning based on user feedback and may have to add non-textual content (sender reputation, IP analysis, link analysis) in order to continue to perform effectively against an evolving threat landscape. The path of research is transitioning away from a "winner-take-all" result and towards an integrated, systems-thinking solution to email security.

6. Conclusion

We have presented a detailed overview of various Machine Learning and Deep Learning techniques used by researchers for spam mail classification and attempted to fuse so many researches together into one view so that the readers can understand how this field has evolved over the years. The investigation is drawing a connection from the early rule based systems to the state-of-the-art of today with complex, context-sensitive models.

The main message of this analysis affirms the trend: in the era where traditional machine learning models such as Random Forest and SVM achieve great success on tabular datasets with handcrafted feature engineering, they are no competitive anymore against deep learning models when it comes to classifying raw, unstructured emails. Specifically, we show that Transformer-based models (e.g., BERT) exhibit superior word accuracy, precision and recall throughout several hard real-world corpora such as Enron-Spam and SpamAssassin. This advantage owes to their capability to learn deep contextual inferences from the text, which is vital for recognizing ironic and spam hidden intents. However, this results in high computational complexity - a crucial trade-off of practical system.

The battle to stop spam is not done. The competing nature of the problem guarantees that new challenges will arise. As we enter an era of concept drift, adversarial attacks and the grim prospect of larger language model-generated highly convincing spam perturbation-a new generation of spam filters is called for-accurate, cloud-based adaptive filtering that can continue to perform despite tampering. The future of spam detection that really works may not turn out to be some single monolithic model, but rather a hybrid, multi-layered system that takes advantage of the strengths of several different algorithmic strategies and which can keep learning on an intraday basis. Ceaseless R&D and continual innovation are the keys to being ahead in this perpetual war for the digital communications protection and security.

7. References

1. Gibson S, Issac B, Zhang L, et al. Detecting Spam Email with Machine Learning Optimized with Bio-Inspired Metaheuristic Algorithms. IEEE Access, 2020;8: 187914-187932.
2. Bhuiyan A, Ashiquzzaman A, Juthi TI, et al. A Survey of Existing E-mail Spam Filtering Methods Considering Machine Learning Techniques. Proc 2018 Int Conf on Innovations in Science, Engineering and Technology (ICISET), 2018: 1-6.
3. Sebastiani F. Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR), 2002;34: 1-47.
4. Stringhini G, Kruegel C, Vigna G. Detecting Spammers on Social Networks. Proc 26th Annual Computer Security Applications Conf, 2010: 1-9.
5. Androutsopoulos I, Paliouras G, Karkaletsis V. Learning to Filter Spam E-mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. Workshop on Machine Learning and Textual Information Access, 4th Hellenic-European Conf on Computer Mathematics and its Applications, 2000.
6. Cormack GV. Email Spam Filtering: A Systematic Review. Foundations and Trends in Information Retrieval, 2008;1: 335-455.
7. Guzella L, Caminhas WM. A Review of Machine Learning Approaches to Spam Filtering. Expert Systems with Applications, 2009;36: 10206-10222.
8. Feng F, Sun D, Zhang L, et al. A New Solution for Spam Filtering. Proc 2016 IEEE Int Conf on Big Data and Smart Computing (BigComp), 2016: 229-236.
9. Devlin J, Chang MW, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.
10. Awad WA, Elseuofi SM. Machine Learning Methods for Spam E-mail Classification. Int J of Computer Science and Information Technology (IJCSIT), 2011;3: 173-184.
11. Rathi S, Parekh R. Spam Email Detection Through Data Mining-A Comparative Performance Analysis. Int J of Modern Education and Computer Science, 2013;5: 31-38.
12. Bhuiyan H, Ashiquzzaman A, Juthi TI, et al. A Survey of Existing E-mail Spam Filtering Methods Considering Machine Learning Techniques. Int J of Computer Science and Network Security (IJCSNS), 2018;18: 7-15.

13. Hopkins M, Reeber E, Forman G, et al. Spambase Data Set. UCI Machine Learning Repository, 1999.

14. Metsis V androutsopoulos I, Paliouras G. Spam Filtering with Naive Bayes-Which Naive Bayes? Proc. CEAS 2006 - Third Conf. on Email and Anti-Spam, Mountain View, 2006.

15. Tusher EH, Ismail MA, Raffei AFM. Email Spam Classification Based on Deep Learning Methods: A Review. Iraqi Journal for Computer Science and Mathematics, 2025;6.

16. Kim Y. Convolutional Neural Networks for Sentence Classification. Proc 2014 Conf on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1746-1751.

17. Vaswani A, Shazeer N, Parmar an, et al. Attention is All You Need. Advances in Neural Information Processing Systems (NeurIPS), 2017;30.

18. Hassan S, Mtetwa N. A Review of Spam Detection Techniques. Proc 2018 Conf on Information Communications Technology and Society (ICTAS), 2018: 1-6.

19. Cranor LF, LaMacchia BA. Spam! Communications of the ACM, 1998;41: 74-83.

20. Almeida TA, Gómez Hidalgo JM, Yamakami A. Contributions to the Study of SMS Spam Filtering: New Collection and Results. Proc 2011 ACM Symposium on Document Engineering, Mountain View, 2011: 259-262.

21. Rissanen J. Modelling by Shortest Data Description. Automatica, 1978;14: 465-471.

22. Postel JB. Simple Mail Transfer Protocol. RFC 1982;821.

23. Crocker D. Internet Mail Architecture. RFC 2009;5598.

24. Wibisono A. Filtering Spam Email Menggunakan Metode Naive Bayes. Jurnal Teknologi Pintar, 2023;3.

25. Akinyelu AA, Adewumi AO. Spam Filtering Using Hybrid of Random Forest and Particle Swarm Optimization. Proc 2014 Int Conf on Computing, Networking and Informatics (ICCNI), 2014: 1-7.

26. Murugavel S, Santhi V. A Density-Based Clustering Approach for Email Classification Using KNN Algorithm. Proc 2020 Int Conf on Computer Communication and Informatics (ICCCI), 2020: 1-5.

27. Mohammad S. A Review on Spam Email Filtering Techniques. Int J of Advanced Computer Science and Applications, 2020;11.

28. Ferreira AA, Oliveira D, Kuehne U, et al. A Review of Spam Filtering Techniques. Applied Soft Computing, 2021;104: 107190.

29. Kaddoura S, Alfandi O, Dahmani G. A Comprehensive Survey on Email Spam Detection," Journal of Network and Computer Applications, 2020;160: 102621.

30. Alqatawna J, Faris H, Jaradat K, et al. Improving Knowledge-Based Spam Detection Methods: The Effect of Malicious-Related Features in Imbalanced Data Distribution. Int J of Communications, Network and System Sciences, 2015;8: 159-172.