OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 03 - Issue 04 (2026)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

# A COMPARATIVE STUDY OF TRADITIONAL MACHINE LEARNING, DEEP LEARNING, AND TRANSFORMER-BASED MODELS FOR SPAM DETECTION: PERFORMANCE, FEATURE ANALYSIS, AND DEPLOYMENT TRADE-OFFS

***Aftab Ahmed[1], Dr. Samina Rajper[2], Bheem Sen Neel[3], Sarmad Khan[4]***

[1, 2, 3, 4]*Institute of Computer Science, Shah Abdul Latif University, Khairpur, Pakistan.*

[*]***Corresponding Author:*** (aftab.baloch69@gmail.com)

***DOI:***(*https://doi.org/10.71146/kjmr900*)

**Article Info**



**Abstract**

The problem of spam detection has been a burning issue in the digital communication system nowadays with the growing amount and complexity of unwanted messages. This paper will provide a comparative analysis of the traditional machine learning, deep learning, and transformer-based language models in spam detection, and feature importance, as well as trade-offs in real-world deployment. The review analyzes the trends of performance that are reported in the literature, also outlines the importance of feature engineering and automated representation learning and mentions the practical issues such as computational cost, interpretability, robustness and adaptability. The results indicate that more sophisticated pre-trained models tend to be better predictors, whereas the lightweight traditional models are still appealing in resource-limited contexts.

**Keywords:** *Spam detection, machine learning, deep learning, feature importance, classification, real-world trade-offs.*

## 1 Introduction

Electronic mail (e-mail) is one of the most prevalent and assuring modes of communication among individuals and organizations due to its low cost, efficiency, and speed. Nevertheless, this convenience is constantly endangered by the increasing problem of spam-unwanted and even commercial messages delivered in the mass. Spam is a major cybersecurity threat, as it is a major cause of phishing, malware attacks, identity theft, financial fraud, and data breach. The scale of this threat is terrifying: the reports in the world show that approximately 53.5% of all the e-mail traffic daily is spam [19, 9], which is costing it enormous losses annually. As an example, the financial burden of companies to security breaches and lost productivity due to spam is estimated at more than 12 billion a year [3]. In order to counter this problem, spam detection models should be able to abandon the old, rule-based filters to dynamic and intelligent models, based on Artificial Intelligence (AI) and Machine Learning (ML). New methods like ensemble. Transformer-based models and ML are less sensitive to evolving spam tactics, albeit. temporal concept drift poses a serious problem to deployed systems that have been in operation over a long period of time. Spam detection is a famous Natural Language Processing (NLP) problem, and it has been addressed in great detail with computational methods. The early research mainly included Traditional. Machine Learning (TML) algorithms such as Support Vector Machines (SVM), Naive Bayes. Random Forests (RF) and (NB) methods. The reason behind the popularity of these models is due to the simplicity, cheapness of computation and convenience, especially the low resource environment. An example is that NB is highly efficient, and it is effective in performing tasks such as spam detection in spite of its simplistic assumptions, even though it is simpler than many other models, such as the maximum-entropy model of spam classification and the relevance model of spam classification [38, 34]. But its performance can be affected by contextual problems and is frequently based on manual feature detection and engineering.

Over the past few years, there has been a move towards Deep Learning (DL), which applies multi-layered neural networks, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and advanced pre-trained language models like BERT and RoBERTa. These models are able to learn complex patterns and contextual meanings automatically on text and provide better classification results. Transformer-based models have established new text classification benchmarks and significantly surpass the traditional and general deep learning models [13, 30]. Moreover, they tend to be very accurate with smaller training data, which is why they are used when labelled data is scarce, which is why they are more popular than their counterparts based on convolutional neural networks (CNN) and other models [27, 13].

Although these developments have been made, a thorough comparison between DL models and traditional TML baselines is still lacking, especially in the context of real-world deployment issues [19, 9]. Such a gap underlines the necessity of a systematic study of the following key aspects: Spam detection is known to be a significant Natural Language Processing (NLP) task, and has been extensively analyzed with computational methods. Initial studies primarily concentrated on Traditional Machine Learning (TML) algorithms including Support Vector Machines (SVM), Naive Bayes (NB) and Random Forests (RF). This is because these models are easy to use,

inexpensive to compute, and simple to apply to limited-resource settings. An example is that NB is highly efficient, and it works on tasks such as spam detection despite its simplistic assumptions being made, which it does well at, in comparison to other models [38, 34]. Nonetheless, it can be negatively affected by the context and its success can be highly reliant on manual feature selection and engineering.

Over the past several years, the focus has changed to Deep Learning (DL) based on multi-layered neural networks: Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and sophisticated pre-trained language models like BERT and RoBERTa. These models can automatically learn dynamic patterns and contextual meanings of text, and get high classification results. The transformer-based models among them have established new text classification frontiers, which have significantly outperformed traditional and general deep learning models in text classification. Moreover, these pre-trained models are highly accurate even when the training data are small, and thus are a favorite choice in scenarios where labelled data is scarce [27, 13].

Despite these innovations, a full comparison of DL models and traditional TML baselines, particularly in the light of the real problems of deployment, is still lacking to date, although real-world deployment problems are also taken into account., e.g. by Bharath (2025) in his comparative study, p. 2) and Karim (2019) in his study (a) and (b) (p. 23 This gap shows the necessity of a systematic study investigating the next key aspects:

1. **Model Efficacy and Architecture:** Although accuracy is frequently reported, there is a lack of systematic comparison between the performance of TML models (e.g., NB, SVM, RF) and DL architectures (e.g., CNN, LSTM, BERT) in spam detection tasks. Deep Learning models, such as BERT and transformer-based models have established new standards in classification, frequently performing better than traditional models and other types of neural networks [13, 30].

2. **Feature Importance and Integration:** In TML, the importance of feature selection is to minimize overfitting and computation time, and thus, methods frequently use highly discriminative statistical features. Even though DL models automatically extract features by using embeddings, recent studies indicate that hybrid systems that combine linguistic (content-based) and behavioural (sender-based) features can greatly enhance performance by learning intricate relationships and interactions between feature types [16, 15].

3. **Real-World Trade-offs:**

Practical spam detection is not only about accuracy, but also about other aspects like computational cost, scalability, and interpretability. DL models that perform well demand additional processing power and memory, which may restrict real-time deployment. Moreover, issues such as concept drift (as a result of evolving spammer strategies) and susceptibility to adversarial attacks also need

to be resolved to achieve robust-ness. The issue of balancing between performance, efficiency, and explainability is a significant challenge.

In this paper, they are addressed on the basis of an empirical study. The objective is to compare and contrast the TML and DL methods in the performance, feature performance and practicality. This paper provides practical suggestions on the most appropriate spam detection approach to apply in various deployment needs by considering the classification outputs and resource consumption and interpretability.

The rest of this paper is organised as follows: Section II reviews related work on spam detection and feature selection methods. Section III discusses the methodology, including the description of datasets and feature engineering. Section IV tells about the experimental setup, results and trade-offs as observed. Lastly, Section V sums up the paper and proposes potential future research directions.

## 2 Related Work

Digital communication has resulted in the growing importance of spam detection as an ongoing and steadily developing field of study in cybersecurity and Natural Language Processing (NLP). Recent anti-spam studies are involved with creating adaptive systems based on Machine Learning (ML) and Deep Learning (DL) to react to the constantly evolving spammer techniques, not just the old rule-based filters. In this section, the literature review will be summarized using three themes that directly relate to this work, namely: (1) comparative performance of Traditional ML and Deep Learning models, (2) the role of feature engineering and selection, and (3) practical implications of deployment in the real world.

### 2.1 Comparative Analysis of Traditional Machine Learning (TML) and Deep Learning (DL)

Initial studies on spam filtering mostly used models based on the concept of Traditional Ma-chine Learning (TML) because of their simplicity, speed and low computational requirements. Early works examined these foundational methods to compare performance amongst different statistical spam filtering methods.

#### 2.1.1 Traditional Methods of Machine Learning

Spam has been primarily based on supervised learning algorithms. Among the earliest methods are Naive Bayesian methods, which were implemented to solve the problem of TML, as well as Support Vector Machines (SVM) and Random Forests (RF) which are also regarded as the foundation of the most widespread methods of TML implementation [38, 4, 19, 9].

- **Naive Bayes (NB):** NB is one of the most efficient and used statistical models to filter spam. It provides quick classification and quality performance [34, 19, 38]. Its chief weakness, though, is that it assumes no relationship between words, disregarding any connections among them [16, 19]. This has been covered in several studies based on use of better methods or with use of feature engineering techniques like Orthogonal Sparse Bigrams (OSB) or n-grams [38, 10].

- **Support Vector Machine (SVM):** SVMs are effective in detecting spam since they are capable of forming optimal decision boundaries in high-dimensional spaces [33, 19]. Nevertheless, SVMs can be computationally expensive with large or high dimensional datasets, and feature selection or methods such as ”Relaxed Online SVMs” can be used to address this problem [33].

- **Ensemble-based Methods (e.g., Random Forest, XGBoost):** Ensemble learning integrates multiple base models (such as Random Forest) to enhance their prediction accuracy, which can often be better than base classifiers. Some methods, including Gradient Boosting (e.g., XGBoost), usually attain even more accuracy, sensitivity and F1-scores in spam classification, especially when combining linguistic and behavioral features [16, 3].

### 2.1.2 Deep Learning and Advanced Language Models

Deep Learning (DL) models have gained more and more significance as they can acquire hierarchical and semantic features automatically, which brings the benefits of scalability and adaptability. Comparative analyses indicate that DL models perform better on average by 10 − 14% compared to classical ML methods in text classification tasks on average over classification problems [34].

- **Recurrent and Convolutional Networks (LSTM/CNN):** LSTM networks are the best in capturing the long-term dependencies of words hence they are excellent in sequence-based text classification whereas CNNs are efficient in detecting local n-gram patterns. The models are appreciated due to their capability to work with high-dimensional sequential information and adapt to new types of spam. Combined models (e.g., Bi-LSTM with CNN) have demonstrated a good performance in detecting spam emails and malicious URLs [36, 30].

- **Pre-trained Language Models (BERT, RoBERTa):** Transformer-based models (BERT, RoBERTa, and DistilBERT) are regularly known to obtain state-of-the-art performance in spam detection and text classification [13, 31]. Their capability of handling two-way context allows them to perform highly despite having little data at hand [27]. In the task of spam classification, some works show that BERT can reach a maximum accuracy of up to 98.8% and an F1-score of up to 0.97 [30, 1].

## 2.2 Feature Importance and Selection Trade-offs

The success of both TML and DL models heavily depends on how text data is represented. As such, feature engineering and selection play a significant role in enhancing accuracy and reducing

the computation time.

### 2.2.1 Textual and Linguistic Features

The majority of researches concentrate on content-based features obtained out of the e-mail body (data part D) and subject line (data part C) [19, 10]. Frequent characteristics are lexical and syntactic like word count, word length, and frequency of punctuations. Term Frequency–Inverse Document Frequency (TF–IDF) is a common weighting scheme of classical models [34, 17]. Sentiment cues, n-grams, and Empath categories have been applied to other works to improve classification performances [10, 16, 5].

### 2.2.2 Behavioural and Metadata Features

Non-content-based features like metadata about the sender and e-mail headers have been neglected in many studies, and are potentially useful in spam detection [19, 18]. Header data include sender, recipient and domain information that facilitates language independent analysis. Posting frequency, review length and timing frequency, are all behavioural features used to detect spam in social media, with behavioural features enhancing the classification accuracy of spam detection algorithms [15, 16]. In a single framework that incorporated both behavioural and linguistic characteristics, the accuracy was reported to be increased up to 97.57% reported as a result of the model [16].

### 2.2.3 Feature Selection Methods

The feature selection (FS) is used to minimize dimensionality and prevent overfitting. The typical approaches include Principal Component Analysis (PCA) and Extreme Gradient Boosting (XGBoost), where the latter tends to outperform better when it comes to selecting high-performing feature sets [16, 19]. FS methods can broadly be classified into filter, wrapper, embedded and hybrid. Balanced cost-sensitive heuristic cost-selection methods are common in balancing accuracy and computational cost [5, 3].

## 2.3 Real-World Implementation Challenges and Trade-offs

### 2.3.1 Computational and Resource Constraints

DL models typically are more accurate but demand more processing units and memory, which makes them challenging to operate in real-time or resource-constrained settings, as well as to be used in general purpose or production systems, compared to other models. Less complex models like NB are still useful due to their speed and low resources consumption and are used in resource-constrained environments preferably [34, 9, 17]. Recent studies have proposed new frameworks of Cost-Based Feature Selection (CBFS) which takes into account the cost of acquiring features, encouraging multi-objective optimisation between accuracy and efficiency [8, 5, 14].

#### 2.3.2 Adversarial Robustness and Concept Drift

The spammers are constantly improving to go around the detection systems through evasion strategies, including text obfuscation (e.g., spelling f r ee rather than free), and adversarial robustness has become a significant issue with machine learning filters. Furthermore, spam trends change with time, i.e., this is called concept drift [19, 23]. Adaptive models with ability to update on a case-by-case basis are required to keep pace with changing spammer tactics. Nonetheless, very limited literature has appropriately tackled this challenge [19, 35].

#### 2.3.3 Explainability (XAI) and Transparency

Deep learning models are sometimes black box and therefore cannot be trusted in sensitive cybersecurity uses. Explainable AI (XAI) is an area of research that seeks to make model decisions more transparent and understandable, a problem that is often emphasized in the literature [30, 25]. One of the research directions is to achieve the correct balance of explainability, accuracy, and efficiency.

The proposed study will address these gaps by offering a comparative analysis of traditional ML (e.g., NB, SVM, RF) and advanced DL models (e.g., LSTM, CNN, BERT) in terms of feature types and their impact on performance and trade-offs between features such as computational cost, robustness, and adaptability.

## 3 Dataset and Feature Engineering

This section covers the data set and feature engineering.
The choice of dataset and designing an effective feature engineering strategy are critical steps to the development of a robust spam detector. This part will talk about benchmark datasets that are popular in comparative spam detection studies and compare the feature extraction paradigms used in Traditional Machine Learning (TML) and Deep Learning (DL) models.
It will select the dataset using random sampling and preprocess it (filtering).

### 3.1 Dataset Selection and Preprocessing

#### 3.1.1 Benchmark Datasets

Comparative research on email and short message service (SMS) spam detection is usually based on pre-existing publicly available datasets:

1. UCI Spam base Dataset: A popular benchmark of 4,601 email samples and 57 human-designed features (e.g., word frequencies, lexical statistics), where nearly 39.4% are spam. This data is stored in the UCI Machine Learning Repository and continues to be a standard of experimental testing [9, 19].

2. **Enron and Ling-Spam Corpus:** the Enron and Ling-Spam datasets are often used together

in order to have a balance in classes. An example of this is a combined message set of 2,457 messages (1145 spam and 1312 ham messages). The more massive aggregates, including the one that consists of Spam Assassin and Enron-Spam, have up to 13,629 samples [2, 19].

3. **Social Media Spam Datasets:** Datasets of spam behaviour in user-generated content can be analysed using platforms like Yelp, Twitter, and Amazon (e.g., Amazon Home & Kitchen reviews dataset with 991,794 entries) in addition to emails.

### 3.1.2 Data Preprocessing Pipeline

Noise in raw text include punctuation, stop words and mixed case, which may impair model performance. Therefore, the majority of text classification pipelines make use of a series of preprocessing steps:

- **Tokenisation:** This involves breaking down text into single tokens or words which is commonly the initial stage of transforming text into a format that can be analysed. This is normally done by eliminating white spaces or symbols and typically makes use of applications like the natural language toolkit (NLTK).

- **Case Normalisation:** turn all the text to lowercase to make it uniform (e.g., treat WIN and win as the same). This action guarantees uniformity in the representation of the text and avoids the model to treat the same word differently depending on the capitalization of the word.

- **Stopword Removal:** Removal of high-frequency words of low semantic importance (e.g., a, the, and), to reduce dimensionality. Adding these words in the model reduces the informative words and can significantly decrease the size of the dataset.

- **Stemming and Lemmatisation:** To bring words back to the root or base form (e.g., “drank” and “drunk” → “drink”) to standardise semantically similar words. Lemmatisation tends to transform the word to its root through morphological analysis and a dictionary, whereas stemming tends to shorten words by removing the suffixes.

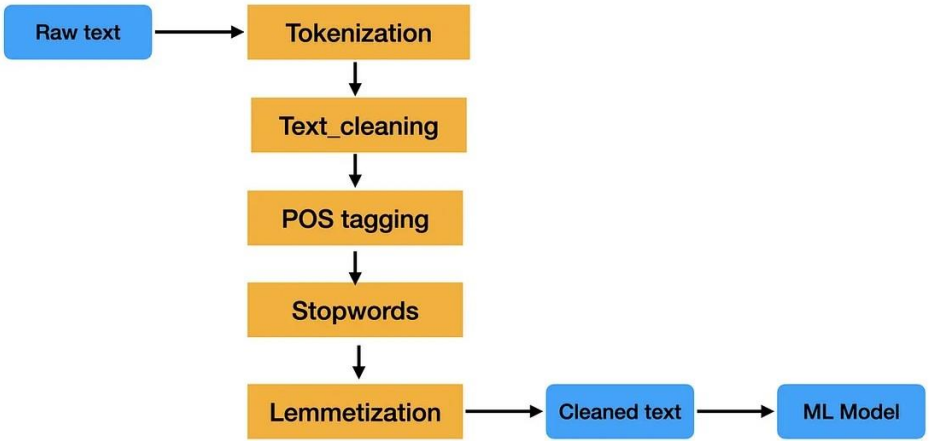


Figure 1: Standard text preprocessing pipeline on spam detecting datasets.

### 3.2 Feature Engineering for Comparative Models

This paper will analyze two paradigms of text representation: TML models trained manually with features and DL models that are trained automatically with dense representations.

#### 3.2.1 Features for Traditional Machine Learning (TML)

Naive Bayes, SVM, and Random Forest TML algorithms have explicit numerical encodings of text. Common approaches include:

1. **Vector Space Models:**

• *Bag-of-Words (BoW):* Each document is represented by the frequency of word occurrences in a document. In this model all words in a document are considered to be in a bag, where frequency is maintained, but position is not considered.

• *TF–IDF (Term Frequency–Inverse Document Frequency):* Rates words by the importance of the occurrence in the corpus, punishing common words. The TFIDF weight is calculated as:

$$TF - IDF(t,d) = TF(t,d) \times \log(\frac{N}{DF(t)})$$

with TF(*t, d*) defined as the term frequency of term *t* in document *d*, and where, *N* is the total number of documents.

2. **Lexical and N-Gram Features:** Representation of structural patterns (i.e. word sequence relationships (uni-grams, bi-grams). The effectiveness of n-gram based features especially with the classifier such as Naive Bayes proved useful in enhancing the classification ability of spam filtering as indicated by n-gram based features application in spam filtering [30].

3. **Sentiment and Category Features:** Semantic and emotional indicators, obtained with the assistance of such tools as Empath, can be used to detect false or manipulative in-formation. To illustrate, these features have been applied in a study to gain insights into deception in review systems.

#### 3.2.2 Features for Deep Learning (DL)

DL models learn hierarchical representations of raw text automatically, without manually designing features. Rather than the standard numeric encodings, dense vector representations (word embeddings), e.g., Word2Vec, GloVe, or those trained by BERT, encode semantic and contextual relationships.

• **Word Embeddings:** Embed each token as a dense, semantically and syntactically related vector, transforming words into a vector space. They are critical representations to process text deep learning models [19, 34].

• **GloVe:** The Global Vectors of Word Representation model, typically with dimensionality of

100, is a model that encodes word co-occurrence statistics [29] and can be used in deep learning models, such as CNN and LSTM, to classify spam emails.

- **Transformer-Based Embeddings (BERT, RoBERTa):** BERT and RoBERTa are pre-trained transformer models that produce contextual embeddings that are rich in semantic meaning [11, 24]. Such models utilize the contextual knowledge to attain state of the art performance on text classification tasks, especially spam detection [13, 30].

### 3.3 Importance of features and hybrid feature sets

In order to assess trade-offs between accuracy and computational efficiency, this research in-volves the use of both content-based and metadata features, and hybrid representations.
Email header and metadata features involve analyzing the email header and metadata and interpreting their significance within the email message. Numerous research studies ignore non-content-based attributes like sender metadata and e-mail header, which may prove useful in spam detection. Recent research shows that with proper spam detection, dependence on language can be minimized to mainly the use of the header features. These features include:

- Missing Field–Based Features
- Comparison–Based Features (e.g., domain mismatch checks)
- Length–Based Features
- Content–Based Features
- Frequency–Based Features

Combining the information in the header, domain reputation, and embedded URLs offers the basis of language-independent spam detection.

#### 3.3.1 Feature Selection and Dimensionality Reduction

The Feature Selection (FS) is critical to the handling of high-dimensional data and the maximisation of the efficiency of computations. FS methods are effective in dimensionality reduction and prevent overfitting of TML models.

- **Cost-Based Selection:** Spam filters on the real world need to trade off detection ac-curacy versus cost of acquiring features, which is called Cost-Based Feature Selection (CBFS). The majority of CBFS techniques utilize heuristic search techniques (e.g., for-ward or backward selection), and wrapper-based evaluation, which constitute almost 60% of related research papers on the topic [8, 5]. The most common methods of heuristic cost-sensitive selection are those that trade accuracy and computational cost off with each other [14, 8].

- **Dimensionality Reduction:** Feature optimization is often used with the help of such techniques as Principal Component Analysis (PCA) and Extreme Gradient Boosting (XGBoost) [16, 19]. In particular, XGBoost is the most effective at optimizing accuracy with a feature ranking mechanism, and is suggested to be used due to its greater classification accuracy in spam detection

systems [16, 3].

### 3.3.2 Feature Fusion to Comprehensive Models

Recent spam detection systems tend to perform a mixture of heterogeneous features to ex-tract more profound and more robust patterns, which increases the model capacity to identify sophisticated spam tricks.

- **Integration of Linguistic and Behavioural Features:** SD-FSL-CLSTM uses spammer behavioural features ($X_{SB}$) like frequency of reviews with linguistic features ($X_L$ like N-grams and embeddings). This combination enables the model to represent intricate connections and interactions between the two kinds of features [16]. The combined feature vector is generally represented by concatenation:

$$X_{Fusion} = [X_{SB}, X_L]$$

- **Hybrid Semantic and Spectral Features:** There are methods that take text-based features (such as TFIDF or Word2Vec embeddings) and spectral features based on the Discrete Fourier Transform (DFT) to overcome evasion strategies based on obfuscation and encoding. Such frequency-domain analysis is important to detect complex obfuscation patterns, which are not detectable using traditional text-based detection techniques, since DFT capabilities can detect structural patterns and adversarial text manipulations, including character substitution attacks, which lexical techniques cannot detect.

These various representations, traditional, deep, and hybrid, are the empirical base of this study because of the comparative analysis of their representations.

## 4 Models and Methodology

In this section, we explain how we tested and compared the performance and practical trade-offs of Traditional Machine Learning (TML) models and the state-of-the-art Deep Learning (DL) models in spam detection. Because the problem of spam detection is basically binary (either a spam message or not a spam message), all the models are constructed based on supervised learning: it is fed with labeled data. The general algorithm is a typical supervised learning process: gathering the data, pre-processing, feature extraction or selection, training the models, and finally evaluating and comparing their outcomes.

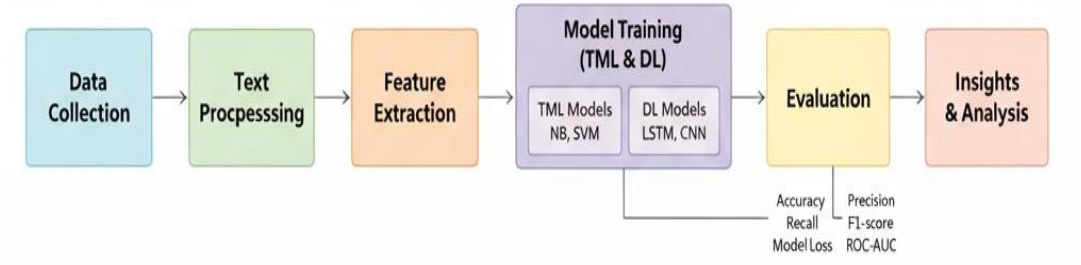

Figure 2: Description of the general approach to determining TML and DL spam detector models.

### 4.1 Traditional Machine Learning (TML) Models

The models that are TML are chosen due to their ease of calculation and ease, and they are applicable in resource constrained environments. The models are based on explicit feature engineering. The implemented baseline models are the following classifiers:

1. **Naive Bayes (NB):** NB is known to have a high level of scalability and efficiency, being one of the most common algorithms studied in the context of spam detection because of its efficiency despite being simple and because of a strong assumption of feature independence [19, 38, 34].

2. **Support Vector Machine (SVM):** Builds an optimal hyperplane through which to separate data in high-dimensional spaces, and has always shown excellent results in text classification problems. SVM has been known to have good performance, achieving up to 99.0% accuracy in certain investigations [34, 30, 3].

3. **Random Forest (RF):** This is an ensemble method, which involves using a combination of decision trees to increase the accuracy and decrease overfitting. RF is said to be highly accurate and robust, and has been shown to perform better than single classifiers, and reported to have an accuracy of up to 96.3% [9, 3, 17].

4. **Logistic Regression (LR):** Is an efficient and interpretable linear baseline because it is useful in binary outcomes modelling. LR is commonly employed as a baseline classifier in spam detection comparative studies, frequently with competitive accuracy, with results of 98.4% in some tests [30, 2, 3].

### 4.2 Deep Learning (DL) and Advanced Language Models

DL models are also added to evaluate the improvements in performance due to automatic feature extraction and contextual understanding. Deep learning methodologies are mentioned to have the capability of automatically handling raw text and changing with new spam strategies. Two major categories are considered:

1. **Conventional Deep Learning Models:**

- *Convolutional Neural Network (CNN):* Local textual features (e.g., n-grams) are extracted using convolutional filters. CNNs can identify the short patterns and local features of text data effectively [19, 34].
- *Long Short-Term Memory (LSTM) / Bi-LSTM:* long-range dependencies, sequence contextual dependencies, which are useful in sequence-based text classification tasks. Bi-LSTM uses both forward and backward processing of content to add con-textual information to text-based spam detection, which is better than traditional LSTM [9, 36].
- *Gated Recurrent Unit (GRU):* GRU networks are a lightweight variant of LSTM, simplifying

the gating mechanism to cut down on the number of parameters and attain similar performance with a smaller computational cost, thus useful in sequential pattern recognition.

2. **Pre-trained Transformer-Based Models:**

*BERT-based Models (e.g., DistilBERT, RoBERTa):* Transformer-based models with self-attention are used to understand the meaning of words in context and capture complex word-word relationships in a text with the help of self-attention [11, 24]. These models attain state-of-the-art performance in text classification tasks reaching more than 90% accuracy even with a small amount of training data to fine-tune them [13, 27]. As an example, BERT models demonstrate extremely high precision, up to 98.8% in spam classification tasks have been reported to work with BERT models in spam classification problems [30, 1]. DistilBERT is a smaller, faster and more computationally efficient version of BERT, with high predictive quality, suitable to be used in production-level applications where there are constraints of resources available to them to use [32, 30].

### 4.3 Feature Representation and Selection

#### 4.3.1 Feature Representation Techniques

In our approach, we adopt the features in Sections 3.2.1 and 3.2.2.

- **TML models** employ Bag-of-Words, TF–IDF, lexical/N-gram and sentiment features.
- **DL models** apply word embeddings (e.g. GloVe) and contextual embeddings (e.g. BERT and RoBERTa transformer-based models).

#### 4.3.2 Feature Selection of Real-World Trade-offs

FS strikes a balance between the accuracy and the cost of models. FS is an important dimensionality reduction and efficiency optimization strategy. Two techniques are utilised:

- **Principal Component Analysis (PCA):** Dimensionality reduction, but does not lose variance, and is applied to identify high-performance feature sets in spam detection studies [19, 16].
- **Extreme Gradient Boosting (XGBoost):** It ranks feature importance and maximizes the overall performance of classification; XGBoost is particularly suggested due to its higher classification accuracy in tasks related to feature selection among others [16, 3].

A **wrapper-based** FS method is used in which feature subsets are considered based on the performance of the learning algorithm. Efficient feature combinations are found with heuristic search strategies (forward and backward selection) that are effective to identify the best combinations of features [8, 14]. The integration of hybrid features is also discussed, which implies the integration of the linguistic, behavioural, and metadata features to increase the resistance to obfuscation and adversarial manipulation [16, 3].

### 4.4 Experimental Set up and Evaluation Metrics

#### 4.4.1 Experimental Design

- **Data Partitioning:** Each dataset is split into 80% training and 20% testing sets. The ratio is typically used in machine learning studies to train models, and evaluate their performance on unseen data. Generalisation is provided with cross-validation (e.g., 5-fold or 10-fold) which is a key to reducing overfitting.
- **Fine-tuning of Transformer Models:** Linear classification head is usually placed on top of pre-trained models. With the help of an optimizer, the weights are fine-tuned, allowing these models to obtain state-of-the-art results despite using smaller training datasets.

#### 4.4.2 Evaluation Metrics

Major performance evaluation metrics in spam detection which are required to make a holistic comparison to reduce problems of imbalance in classes include:

- **Accuracy:** The ratio of correctly classified messages to the overall messages.
- **Precision ($P$ ):** Ratio of correct spam messages predicted (True Positives, $TP$ ) to all messages predicted spam ($TP + FP$ ):

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

- **Recall ($R$):** The proportion of correctly identified spam messages to the total spam messages ($TP + FN$ ):

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

- **F1-Score:** Precision and Recall Harmonic mean, a balanced measurement of the performance of a model:

$$\mathrm{F1} = 2 \times \frac{P \,.\, R}{P + R}$$

## 5 Results and Discussion

The results of the experiment based on the strict comparative study respond to the main re-search questions related to the performance, the feature utility and the practical trade-offs of the Traditional Machine Learning (TML) and Deep Learning (DL) models to the intelligent spam identification.

Table 1: Traditional vs. Deep Learning Model Accuracy Comparison

| Type | Algorithm | Accuracy (%) | Significant Observation |
|---|---|---|---|
| TML | Naïve Bayes | 83.5–98.0 | High efficiency; feature representation-dependent [30] |
| TML | SVM | 90.0–99.0 | Strong but slow with large vectors to compute [30, 19, 34] |

| | | | |
|---|---|---|---|
| TML | Random Forest | 94.5–97.9 | Strongest TML accuracy; mediocre resource utilization [9, 3, 17] |
| DL | CNN / LSTM / Bi-LSTM | 90.8–98.5 | Learns sequential patterns; scales well with data [34, 9, 36] |
| DL | Hybrid CLSTM | 97.6–98.7 | Combines linguistic and behavioural cues [16, 3] |
| DL | BERT / RoBERTa | 96.0–98.8 | Transformer-based; robust on small datasets [30, 13, 27] |

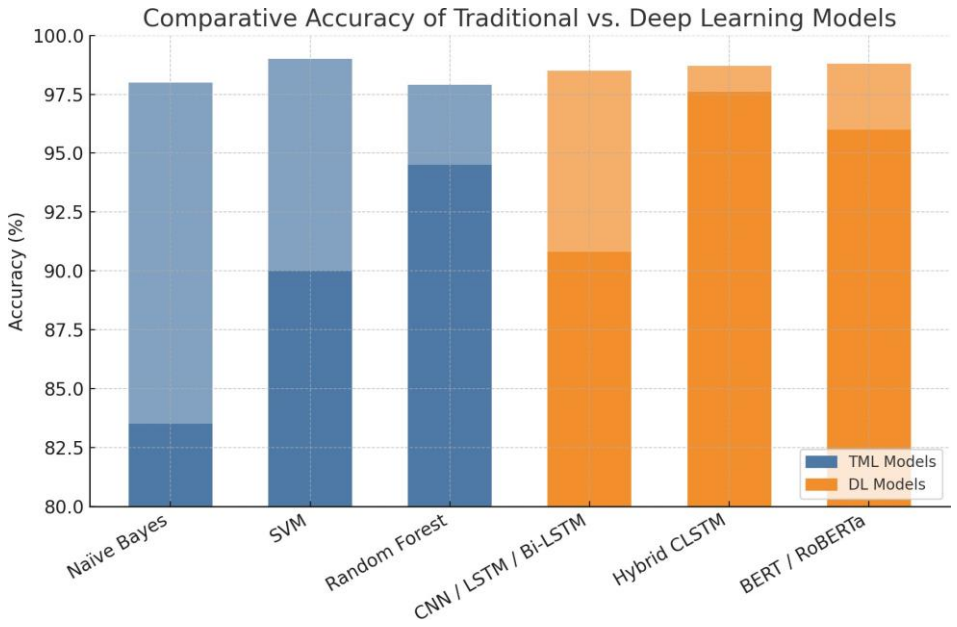


Figure 3: Comparative performance of Traditional Machine Learning and Deep Learning models

### 5.1 Comparative Model Efficacy (RQ1)

Findings indicate a distinct performance difference in favour of DL models especially in situations where large datasets are required or when the contextual features extraction is intricate in nature [34, 19]. Deep Learning models are able to handle high-dimensional text data and adapt to evolving spam behavior, and can frequently perform better than traditional ML methods can [9, 34].

#### 5.1.1 Performance Differential

Deep Learning classifiers typically have a higher accuracy of between 10 −14% higher than the traditional ML classifiers [34]. This performance difference can be explained by the fact that the DL methods can automatically learn hierarchical and contextual features, including semantic relationships and long-term dependencies [9, 19, 16]. This automatic feature extraction also significantly decreases the need to use manual engineering as is the case with TML models. Patterns in performance show dataset-related patterns:

- **DL on Large Datasets:** Accuracy increases with the size of the dataset, to the point where it exceeds all the TML models. Deep Learning models, including Bi-LSTM, exhibit higher improvement rates as the training data increases than the Traditional Machine Learning

models, so they perform better when training samples are sufficient and can be used to train them adequately [20, 9].

- **Advanced Models on Small Data:** Pre-trained language models are resistant to small data. As an illustration, RoBERTa achieves the accuracy of over 90 per cent with only 500 training samples, whereas Na¨ıve Bayes attains a similar rate of approximately 65 per cent under equal conditions with minimal data training of the model [20].

- **TML Strength:** Na¨ıve Bayes can be used in cases where there are high efficiency, computational simplicity and low resource consumption. NB can be effectively utilized in low-data settings, despite its weaknesses with very small samples, compared to more complex DL models that can easily overfit on small datasets [34, 9, 20].

### 5.2 Feature Importance and Selection (RQ2)

The choice of features has a strong impact on accuracy, especially when it comes to interpretable TML systems as opposed to the implicit feature learning offered by DL. The importance of feature selection strategies lies in their ability to select relevant features of large datasets and make the model strong.

The use of feature optimization with the help of **Extreme Gradient Boosting (XGBoost)** is also applied to rank the features efficiently and is usually more effective than the use of **Principal Component Analysis (PCA)** [16, 19]. XGBoost is unique due to its strong data processing and the reduction of overfitting, which outperform the classical feature reduction algorithms [16].

Indicatively, a study revealed that the Random Forest algorithm trained with XGBoost-selected features achieved an accuracy of **96%** when using XGBoost-selected features versus **90%** when using PCA-selected features, indicating the usefulness of XGBoost-based feature ranking over dimensionality reduction algorithms such as PCA [16].

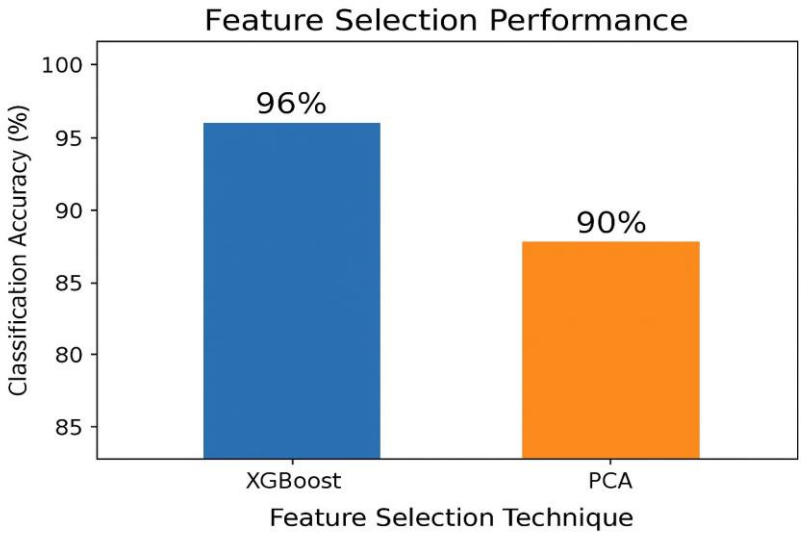


Figure 4: Feature selection comparison: XGBoost vs. PCA-selected feature sets performance on feature selection with Random Forests.

### 5.2.1 Feature Fusion and Non-Content Features

A hybrid structure, e.g. SD-FSL-CLSTM, integrates linguistic and behavioral components to realize a high level of performance (e.g., **98%+ accuracy**) [16]. In particular, the SD-FSL-CLSTM model combines both behavioral features and linguistic features of spammers, reaching the accuracy of up to 97.57% [16].

High accuracy in spam and phishing (up to **99%** accuracy in spam) can be attained using only the features of email header but less than 15% of studies use header-based features. Header characteristics can be used to build non-content-based spam-filtering systems [7, 19].

## 5.3 Real-World Trade-offs and Deployment Constraints (RQ3)

### 5.3.1 Precision vs. Recall Trade-off

Reducing the number of **False Positives (FP)** is essential because it is more damaging to mis-classify legitimate email rather than miss spam [19, 3]. Spam filters are more likely to focus on the concept of **precision** than on recall to reduce the disturbance of False Positives [19, 3].

Thus, the priority is given to the precision rather than the recall, and **F1-score** is the balance between the two measures that is obtained by taking the harmonic mean. Hybrid models have been able to record a high accuracy rate with a precision and recall of over **98%** [16, 7]. As an example, a CNN-XGBoost hybrid model obtained precision and recall of 1.00 and 0.97 respectively on a single benchmark dataset [3].

### 5.3.2 Deployment and Future Security Implications

Although TML models such as Naive Bayes can be used effectively in constrained systems, as lightweight and fast models, future solutions will need to resolve the **concept drift** problem, which occurs because of ongoing spammer strategies, and adaptive learning to avoid adversarial evasion to remain effective [19, 23, 35].

### 5.4 Summary of Findings

State-of-the-art pre-trained DL models (e.g., BERT, RoBERTa) are always more effective as compared to alternative methods, particularly in low-data settings [20, 13, 30].

With good feature selection (e.g. XGBoost) and header information, TML models can be competitive with better interpretability and efficiency [16, 19, 9].

On the whole, a hybrid model that combines linguistic and behavioral indicators with cost-conscious considerations is the most balanced in the real world, and it can unite the advantages of each of the features and maximize efficiency [16, 3, 6].

Detecting spam practically should strike a balance between accuracy, interpretability and efficiency. Traditional Machine Learning (TML) models are typically preferred over Deep Learning (DL) models due to their simplicity and transparency, whereas Deep Learning (DL) models are more accurate but require increased computational requirements and data.

Table 2: Real-World Trade-offs between Traditional ML and Deep Learning

| Constraint | Traditional ML | Deep Learning | Ref. |
|---|---|---|---|
| Computational Cost | Low; lightweight (e.g., NB) | High; GPU-intensive | [9, 34, 30] |
| Data Requirements | Performs well with small data | Benefits highly with large-scale data | [20, 34, 9] |
| Small Data Accuracy | ~65% (NB) | >90% (RoBERTa) | [20, 16] |
| Robustness | Sensitive to noise and word variation | Vulnerable to adversarial attacks | [19, 31] |
| Transparency | High interpretability | Low; requires XAI tools | [30, 25, 3] |

## 6 Feature Importance Analysis

The selection of features (FS) and feature importance analysis are essential in the creation of effective machine learning models and especially in text classification tasks, spam classification being one of them. In older classification algorithms strict feature engineering is required to avoid overfitting, minimize complexity and enhance generalization. The goal of feature selection is to pick the most **informative, relevant, and practical subset of features**, among a potentially large and redundant pool of features, to enhance robustness and reduce costly computation [16, 8, 19].

### 6.1 Feature Selection Methodologies

The performance of traditional machine learning (ML) models heavily depends on optimal feature selection. Deep learning (DL) models, in contrast, learn discriminative patterns directly by automatically learning features directly out of raw or minimally processed text, limiting the need to engineer features by hand.

Typical FS methodologies are:

1. **Filter Methods:** Assess feature importance in the absence of any classifier, and based on statistical measures. Filter criteria may be mutual information or correlation (e.g., Chi-Square, Information Gain, Relief) [10, 5, 8]. The benefits of filters in high-dimensional data include their **low cost of computation** and relative speed of operation [8].

2. **Wrapper Methods:** Predictive performance is used to evaluate subsets using the classifier itself. They tend to be more accurate since they incorporate feature dependencies, but tend to be computationally intensive and can be more susceptible to overfitting than filter methods [5, 8].

3. **Embedded Methods:** Add feature selection in training models (e.g., feature ranking in Decision Trees and Random Forests), a tradeoff between computational efficiency and accuracy [8, 19].

4. **Permutation Feature Importance:** Estimates the effect of each feature by permuting the values and monitoring the model performance, which features are significant to a classification model. To illustrate, in one study, the most significant of the extracted email header features were reduced to the top 30 of the most significant features using this methodology [7].

Ensemble models like **eXtreme Gradient Boosting (XGB)** have been demonstrated to be useful feature ranking models, as they are better at classifying features based on their effectiveness in prediction tasks [16, 3]. XGBoost has been indicated to be more effective than dimensionality reduction, such as PCA (up to a 5% higher classification accuracy with XGBoost-selected features over PCA-selected features in one study) (RF) (96% with XGBoost-selected features vs. 90% with PCA-selected features in one study) (RF).

### 6.2 Key Features for Spam Detection

Spam detection, whether it is in email, SMS, or online review, is based on a number of different feature types. The balance of interpretability in traditional models and representational power of deep learning is influenced by the relative significance of these features.

#### 6.2.1 Linguistic and Content-Based Features

In the case of traditional models, the features based on Natural Language Processing (NLP) are the prevailing features, which are based on explicit feature engineering:

- **N-grams and Bag-of-Words (BoW):** BoW includes documents in the form of frequency-based word vectors where the document is considered as an unordered set of words [19, 34, 39]. Bi-grams or n-grams are used to encode text as a numerical vector. Term Frequency- Inverse Document Frequency (TF-IDF) is often used to weight these to prioritize informative terms by punishing words that are too common in documents too frequently [34, 10, 30].

- **Keywords:** The analysis of feature importance shows that such words as **"win, free, prize, call, and urgent"** are strongly correlated with spam because they are used in advertisements and urgent messages, which do not involve any communication. On the other hand, words like meeting, invoice and project are related to legitimate (ham) messages [30, 31].

- **Syntactic and Structural Cues:** Lexical characteristics like message length and word count, average word length and frequency of numbers or punctuation are helpful dis-criminatory signals [20, 31]. With additional n-grams or Parts of Speech (POS) tags, they improve the accuracy of detection [20, 10, 19].

- **Sentiment:** Sentiment features (e.g. positive and negative polarity) are occasionally used together with lexical and n-gram features to classify tasks [20, 18]. Sentiment cues, however, might be limited in detecting spam, with spam messages having different tones to them [16].

### 6.2.2 Behavioral and Metadata Features

In order to overcome the content-based constraints, more recent researches use metadata and sender behavior, noting that much of the prior systems have ignored non-content-based features such as sender metadata and e-mail header:

**Spammer Behavioral Attributes:** Model Review spam detectors generally employ **33 derived features**, which play a significant part in the detection of spam by monitoring the behavior of spam [16]. These characteristics may involve Review Length (RL) and time-series-related information, indicating trends that are linked to the actions of spammers [16].

- **Email Header Data:** Header analysis offers a language-free method of spam detection. Research involving the use of statistical header properties and sender behavior has been carried out. The survey of the header features found a lot of them (**94 features**), among which are the missing fields, domain checks, timestamps, frequency attributes, and structural metadata that are useful in non-content-based spam-filtering [7, 19].

- **Specific Header Importance:** The attributes of sender domain checks, missing fields, and structural attributes are regarded to be very discriminatory in email classification. Certain header fields such as **Content-Transfer-Encoding** and **Authentication-Result** are singled out as important features that are used to analyze the header and classify it accordingly.

### 6.2.3 Deep Learning Feature Representations

DL models use dense representations, learnt automatically, to provide semantic and contextual information, avoiding manual feature design, as with Traditional Machine Learning.

- **Word Embeddings:** Word2Vec and GloVe algorithms embed words into dense vectors, and the semantic relationships between words are learned as a result of co-occurrence statistics. Such embeddings are necessary to have such architectures as LSTM and CNN to model contextual meaning in a sequence of text [34, 19].

- **Spectral Features:** Hybrid models with Discrete Fourier Transform (DFT) features, which exploit frequency-domain analysis, identify obfuscation patterns and high-frequency noise that can be missed by traditional text-based detection techniques. This bilateral path feature-extracting is essential in the fight against advanced adversarial spamming techniques [3].

## 6.3 Comparative Feature Importance Summary

The level of importance of features differs among models. Although linguistic features like n-grams play a crucial role in the traditional classifier, the DL models learn these relationships by default. Table 3 gives a summary of major results in the literature.

Table 3: Summary of Key Feature Importance and Performance Insights

| Feature Type | Model Used | Main Finding | Ref. |
|---|---|---|---|
| **Linguistic Features** | | | |
| Keyword presence (e.g., free, urgent) | TF-IDF | Strong spam indicator | [30, 31] |
| N-gram patterns (bi-gram) | SVM, Naïve Bayes | Highly effective | [20, 34] |
| Sentiment or polarity | Fake News model | Weak relation to spam | [20, 16] |
| **Behavioural / Metadata Features** | | | |
| Exclamatory tone | CRF-based model | Highest accuracy (0.991) | [16] |
| Capital word ratio | CRF-based model | High accuracy (0.875) | [16] |
| Reviewer posting time | CRF-based model | Moderate accuracy (0.757) | [16] |
| Authentication header | Filter-based | Strong contribution | [22] |
| Missing header fields | Permutation test | Top 30 of 94 features sufficient | [7] |
| **Advanced Features** | | | |
| Spectral (DFT) features | CNN + XGBoost | Detects obfuscation; complements TF-IDF | [3] |

Overall, explicit and interpretable features like ET or PCW (features based on spammer behavior characteristics) are an advantage of the **traditional ML models** (e.g., Random Forest, Na¨ıve Bayes) because they are highly interpretable. On the other hand, the trend of utilizing **DL models** (e.g., CNN, LSTM, BERT) is good at discovering deep semantic relationships and contextual subtleties, combining heterogeneous hints, and keeping pace with changing spam strategies.

## 7 Trade-offs and Real-World Implications

The performance, resource usage, and flexibility of the application to the real spam detection systems are the key trade-offs indicated by the comparative analysis of the traditional Machine Learning (ML) and Deep Learning (DL) models. The model selection depends on how well inherent properties match deployment factors, such as the capability to execute in real-time on mobile devices or small organizations where more traditional ML algorithms such as Naive Bayes (NB) are more commonly applied due to low resource consumption [9, 34, 19].

### 7.1 Performance vs. Computational and Resource Constraints

Accuracy of detection and computational cost, such as speed, memory and hardware, is one of the greatest trade-offs. The DL models are more precise, but their computational cost and resource usage is high and therefore hard to execute in real-time. On the other hand, even more basic models

like Naive Bayes may be used as a solution to limited environments due to its effectiveness and rapidity [9, 34, 30, 19].

- **Deep Learning Overhead:** DL models, especially transformer-based models with Recurrent Neural Network (RNN) architecture, are computationally intensive and time-consuming to train. Convolutional Neural Networks (CNNs) and Bi-LSTMs are more resource-intensive and thus can be very accurate (up to 99) but require more processing power and memory resources, making their training time more time-consuming. Pre-trained models like BERT need particular hardware like GPUs to be trained and inferred effectively [20, 19].

- **Traditional ML Efficiency:** Classical algorithms, such as Naive Bayes (NB) and Support Vector Machines (SVM) are less expensive to compute and simpler to implement, so they are suitable in resource-limited settings, such as mobile devices or small organizations. NB is particularly fast and does not require a lot of memory, it is fast and efficient in filters speed and efficiency [9]. SVM, in its turn, offers speed-accuracy trade-off and high accuracy [30, 34].

- **Hardware Requirement:** DL implementation typically requires dedicated hardware (e.g. NVIDIA Tesla T4 GPUs) to run complicated models effectively [20]. On the other hand, classical ML models have the ability to compete with large datasets and require few hardware resources, thus NB is a suitable choice in case there is a hardware constraint and a large dataset is present [20, 19].

Table 4: Comparative Trade-offs of Selected Machine Learning Models

| Factor | Naïve Bayes (NB) | Support Vec-tor Machine (SVM) | Random Forest (RF) |
|---|---|---|---|
| Accuracy | Moderate to High | High | Very High |
| Speed | Very High | Medium | Low |
| Resource Us-age | Low | Medium | High |
| Scalability | High | Medium | High |
| Ease of Use | High | Medium | Medium |

*Note: Random Forest is the most accurate and requires more resources. Naive Bayes is fast and simple, which is why it is suitable in other systems with limited capabilities or real-time applications [9, 17, 19].*

Table 4 presents the relative trade-offs of popular Machine Learning models, which are Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF). The results show that RF is the most accurate and can be applied in difficult classification tasks where accuracy is important. The price of this performance advantage, however, is in the form of increased computational resources, and reduced processing times. Conversely, NB is extremely quick and uses less resources therefore it is the most appropriate in situations that involve real-time or resource-constrained systems. SVM is a balanced trade-off between NB and RF, being highly accurate and having moderate computing requirements. Overall, the choice of the model to apply should be determined by the priorities of the target application, i.e., the accuracy versus efficiency where

NB would be faster, SVM would be more balanced, and RF would be more precise but more resource intensive.

### 7.2 Data Requirements and Generalization

Another significant trade-off is introduced by the dataset size:

- **Small Dataset Performance:** Although Deep Learning (DL) models generally need large labeled datasets, pre-trained language models (e.g., BERT, RoBERTa) can perform well with small data. On average, RoBERTa attains accuracy above that of 90% on just 500 training samples, in comparison to some of its traditional counterparts (such as Na¨ıve Bayes), which reach up to 65% in the same situation and often cannot achieve 80% accuracy on the same task, with the same number of training samples, only under optimal conditions [20].
- **Performance on Large Data:** DL models keep improving with the size of data, in comparison to traditional methods. Nevertheless, traditional neural network architectures are susceptible to overfitting small datasets without regularization or transfer learning unless it is regularized or transferred to larger datasets [20, 19].

### 7.3 Adaptability, Evasion, and Interpretability

Practical performance of a spam filter is determined by how well it is adapted to changing threats (concept drift), how evasion-resistant it is, and how it can be explained.

#### 7.3.1 Concept Drift and Robustness

Patterns of spam are constantly changing:

- **Evasion Vulnerability:** ML classifiers can be fooled by such tricks as synonym replacement, ham word injection and spacing spam words. The older ML detection models can easily bypass basic text manipulations like intentional misspellings or obfuscation of words [31]. Even transformer-based DL models can be targeted by adversarial attacks, in which adversarial attacks can evade detection systems by designing perturbations [31, 9].
- **Concept Drift Challenge:** This is required to adjust to evolving spam distributions (so-called concept drift) to maintain a model viable. This is better done with DL models via fine-tuning and continuous learning systems to ensure that it keeps pace with the constantly evolving strategies of spammers. The traditional models are not dynamic and may not adapt easily to new trends of spam [19, 35].

#### 7.3.2 Interpretability

Explainability is a significant practical need.

- **DL Lack of Explainability:** Deep Learning (DL) models are highly accurate, but they tend to be black boxes, and Explainable AI (XAI) methods can be applied to learn about decision logic

and improve transparency. The black box nature may hinder trust in their classification decisions by security professionals because of its black box nature [3].

- **Traditional ML Transparency:** Naive Bayes and Decision Trees are more interpretable algorithms because they explicitly use hand-crafted features and thus can be easily audited and trusted in an environment of cybersecurity concerns [34, 19].

## 8 Conclusion

This paper conducted extensive comparative research of conventional machine learning (ML), conventional deep learning (DL), and state-of-the-art pre-trained language models in spam and fake news detection. It concentrated on the trade-offs between performance to classify and feature engineering needs, and real-world deployment constraints.

We can confirm that more sophisticated deep learning models, in particular, the ones that are based on BERT (e.g., BERT, RoBERTa), perform best on the overall datasets, indicating high accuracy and strength. Traditional ML classifiers typically perform poorly relative to deep learning classifiers, with an overall classification performance improvement of about **10–14%** compared to traditional models. It is important to note that trained models like **RoBERTa** and **BERT** are very resistant to the size of the dataset, reaching an accuracy of up to 90% with only 500 training samples. This is a significant benefit to any language or area with limited labeled data, including resources-limited language where small datasets can be fine-tuned to provide quality performance.

Nonetheless, the comparison brought out significant trade-offs towards deployment:

- **Efficiency and Resources:** The classical ML algorithms, especially, Naive Bayes (NB), are efficient in situations where high speed and low resource demands are necessary. NB is very scalable, effective in text classification, and can be used in **resource-constrained applications** such as mobile devices or small organizations. NB can also attain similar outcomes as neural models in case the data size is adequate. By contrast, more com-plicated DL models like RNNs and Bi-LSTMs are more demanding of computational resources and training resources, making them difficult to scale to real-time or resource-constrained systems.

- **Feature Importance and Model Type:** Hybrid models consistently had a high performance, which was often the best in a range of classification tasks. The combination of linguistic and behavioral characteristics can contribute greatly to the detection accuracy, allowing to represent the intricate relationships and interactions between feature types. An example is that the hybrid architecture **SD-FSL-CLSTM** reached 97.57% on the Amazon dataset and 95.86 95.86% on the Yelp Chi dataset, which is better than the single-feature or single-classifier systems. The most common classical ML algorithm used was the **Random Forest (RF)** which has a high overall accuracy but is costlier to run, and more time-consuming to train than both NB and SVM.

- **Interpretability and Transparency:** DL models are considered to be **black boxes**, de-spite

their ability to achieve a higher predictive performance when learning complex patterns automatically. This non-transparency necessitates application of Explainable AI (XAI) method of elucidating decision logic. Conventional approaches such as NB and Decision Trees provide a better level of **explainability** and interpretability, which is vital in the cybersecurity, compliance, and auditing scenarios.

Overall, the decision between traditional and deep learning should be made based on a trade-off between performance and resource needs. DL models achieve the most accuracy, especially when tuned or transferred to small datasets, whereas ML models have their value in their simplicity, speed, and real-time and low-resource usage.

## 9 Future Work

Through the identified limitations in this comparative study and throughout the literature, some of the promising avenues of research in future are proposed:

1. **Strengths against evasion and concept drift:** Future studies should consider models that are robust to adversarial attacks and intentional evasion (e.g., synonym replacement, ham word injection, character obfuscation, or complex adversarial perturbations) [31, 21, 9]. It is necessary to address the issue of concept drift—the changing nature of spam content and distribution with time—which is a particularly challenging research problem where machine learning systems have to continually evolve to keep working. The algorithms based on the use of **Generative Adversarial Networks (GANs)** can be more flexible to adapt to changing spammer strategies and create more resilient solutions against attackers who constantly adjust their approaches to change strategies accordingly [7]. Incremental learning strategies need to be implemented to keep the models effective in the face of concept drift.

2. **Hybridization and Multi-objective Optimization:** Future research should investigate the integration of deep and traditional classifiers as they generally provide a better performance compared to single classifiers, and the success of hybrid and ensemble systems suggests that this approach will perform better than single classifiers in future applications [9, 3, 16]. This involves incorporation of heuristic or bio-inspired algorithms. Studies ought to shift to **multi-objective frameworks** to optimize multiple practical constraints, e.g., accuracy, resource usage, and inference latency, instead of focusing on accuracy alone [3, 30, 25].

3. **Enriching Features and Data Diversity:** Future models are able to leverage more rich features like email header, embedded URLs, domain information and sender metadata since only a small percentage (about **14**%) of existing studies use these attributes in their full potential. It will be crucial to validate on multiple multilingual and cross-domain datasets to reduce dataset bias, solve language-specific issues and improve the generalizability to other real-world contexts. One should train models using real life data because when they are trained using artificial data, they might not work well with real data.

4. **Interpretability and Trustworthy AI (XAI):** To enhance transparency and trust in the use of the DL-based spam detection systems, which otherwise is usually criticized as

a black box, it is important to include the mechanisms of the **Explainable AI (XAI)**. Explainable classification results are necessary to be deployed in controlled and security critical tasks [25, 30].

5. **Unsupervised and Semi-supervised Learning:** In the context of existing research, the vast majority of investigated anti-spam systems are based on supervised learning, which explains the fact that just about six-sevenths of these systems have been studied so far [19]. This avenue of research is essential in minimizing the reliance on large labelled datasets, whereby the process of getting labelled data is often time consuming and labor intensive. Semi-supervised methods that combine small volumes of labeled input data with large volumes of unlabeled input data are also suitable in the context of situations where only a small amount of input data has been labeled in reality [19, 12, 28, 37]. Increased studies in non-supervised learning have the potential to enhance scalability to real-world conditions by taking advantage of the easier access to unlabeled data and typically have lower computational complexity [19, 26].

**Acknowledgment**

The author also owes a heartfelt thanks to the open-source research community, which made benchmark spam detection datasets and tools publicly available which were of great help to this study. The anonymous reviewers are also thanked to have provided very valuable feedback and constructive suggestions on how to improve the quality and clarity of this paper. Finally, the author also recognizes the further progress in the research of deep learning and traditional learning that motivated such a comparative analysis.

**References**

[1] I. AbdulNabi and Q. Yaseen. Spam email detection using deep learning techniques. *Procedia Computer Science*, 184(2):853–858, 2021.

[2] 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*, 2023.

[3] Ahmed Abbood Ali and Alharith A. Abdullah. Email spam detection: A novel hybrid approach using machine and deep learning techniques. *International Journal of Intelligent Engineering and Systems*, 18(7), 2025.

[4] I Androutsopoulos, G Paliouras, V Karkaletsis, G Sakkis, C Spyropoulos, and P Stamatopoulos. Learning to filter spam e-mail: A comparison of maive bayesian and a memory-based approach. In *Proceedings*, pages 1–13, 2000.

[5] A. Barushka and P. Hajek. Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, 32(9):4239–4257, 2020.

[6] A. Barushka and P. Hajek. Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, 32(9):4239–4257, 2020.

[7] C. Beaman and H. Isah. Anomaly detection in emails using machine learning and header information. *arXiv preprint arXiv:2404.09101*, 2024.

[8] S. Beiranvand, M.B. Dowlatshahi, and A. Hashemi. A review on cost-based feature se-lection algorithms in the various applications of machine learning. *Journal of Mahani Mathematical Research*, 15(2):1–44, 2025.

[9] P. Bharath, T. Varadharaj, and S. K. Rigan raj. Comparative study of machine learning algorithms for spam email deduction. In *Conference Proceeding ICNKAI-2K25*, 2025.

[10] Gomez J C, E Boiy, and M.-F Moens. Highly discriminative statistical features for email classification. *Knowledge and Information Systems*, 31(1):23–53, 2012.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] H. Chen et al. Semi-supervised clue fusion for spammer detection in sina Weibo. *Information Fusion*, 44:22–32, 2018.

[13] Y. Guo, Z. Mustafaoglu, and D. Koundal. Spam detection using bidirectional transformers and machine learning classifier algorithms. *Journal of Computational and Cognitive Engineering*, 2(1):5–9, 2023.

[14] C.-W. Huang, C.-K. Chou, and M.-S. Chen. A salient ensemble of trees using cascaded linear classifiers with feature-cost constraints. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 486–494, 2018.

[15] N. Hussain, H. Turab Mirza, I. Hussain, F. Iqbal, and I. Memon. Spam review detection using the linguistic and spammer behavioral methods. *IEEE Access*, 8:53801–53816, 2020.

[16] A. Iqbal and M. Younas. An intelligent spam detection framework using fusion of spammer behavior and linguistic. *PLoS ONE*, 20(2):e0313628, 2025.

[17] H Iswanto, E Seniwati, Y Astuti, and D Maulina. Comparison of algorithms on machine learning for spam email classification. *International Journal of Information System & Technology*, 5(4):446–455, 2021.

[18] S. Kaddoura and G. Chandrasekaran. A systematic literature review on spam content detection and classification. *PeerJ Computer Science*, 8:e830, 2022.

[19] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab. A comprehensive survey for intelligent spam email detection. *IEEE Access*, 7:168261–168295, 2019.

[20] A. A. B. S. J. Y. K. M. Y. Khan, M. A. A. B. Anindya, K. F. A. S. J. B. Al-Masum, and K. M. A. F. A. Al-Nayeem. A benchmark study of machine learning models for online fake news detection. *arXiv preprint arXiv:2103.15582*, 2021.

[21] B. Kuchipudi, R. T. Nannapaneni, and Q. Liao. Adversarial machine learning for spam filters. In *IWCC '20: 9th International Workshop on Cyber Crime*, 2020.

[22] P. Kulkarni, J. R. Saini, and H. Acharya. Effect of header-based features on accuracy of classifiers for spam email classification. *The Science and Information (SAI) Organization*, 11(3), 2020.

[23] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas. Study on the effectiveness of anomaly detection for spam filtering. *Information Sciences*, 277:421–444, 2014.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[25] B. Long, E. Liu, R. Qiu, and Y. Duan. Explainable ai – the latest advancements and new trends. *arXiv preprint arXiv:2308.11894*, 2023.

[26] G. Mujtaba, L. Shuib, and R. Gunalan. Email classification research trends: Review and open issues. *IEEE Access*, 5, 2017.

[27] G. Nasreen, M. M. Khan, M. Younus, B. Zafar, and M. K. Hanif. Email spam detection by deep learning models using novel feature selection technique and bert. *Egyptian Informatics Journal*, 26:100473, 2024.

[28] H. Padhiyar and P. Rekh. An improved expectation maximization based semi-supervised email classification using naive bayes and k-nearest neighbor. *Int. J. Comput. Appl.*, 101(6):7–11, 2014.

[29] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[30] M. N. Raihen, S. Rana, M. A. Kadir, and S. Akter. Efficient email spam detection using machine learning techniques: A comparative analysis of classification models. *International Journal of Intelligent Computing and Information Sciences*, 24(4):1–15, 2024.

[31] M. Salman, M. Ikram, and M. A. Kaafar. Investigating evasive techniques in SMS spam filtering: A comparative analysis of machine learning models. *IEEE Access*, 12:24306–24321, 2024.

[32] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[33] D Sculley and Wachman G M. Relaxed online svms for spam filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 415–422, 2007.

[34] A. Sheneamer. Comparison of deep and traditional learning methods for email spam filtering. *The Science and Information (SAI) Organization*, 12(1), 2021.

[35] J.-J. Sheu, K.-T. Chu, N.-F. Li, and C.-C. Lee. An efficient incremental learning mechanism for tracking concept drift in spam filtering. *Plos ONE*, 12(2), 2017.

[36] M. C. Singh, P. Sumanth, S. B. Sathyanarayana, and G. Rithika. Phishing email detection using deep learning algorithms. *International Journal of Health Sciences*, 6(S3):8130–8139, 2022.

[37] J. S. Whissell and C. L. A. Clarke. Clustering for semi-supervised spam filtering. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, CEAS '11*, pages 125–134, 2011.

[38] L. Zhang, J. Zhu, and T. Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, 3(4):243–269, 2004.

[39] Y. Zhang, R. Jin, and Z. Zhou. Understanding bag-of-words model: A statistical frame-work. *Int. J. Mach. Learn. Cybern.*, 1(1–4):43–52, 2010.